Application of Machine Learning in Warranty Management of Consumer Durables

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Technology and Master of Technology

by

Yash Baley

(13D100034)

Guide

Prof. A. Subash Babu

Department of Mechanical Engineering

Indian Institute of Technology, Bombay

June 2018

Declaration

I declare that this written submission represents my concepts in my very own words and wherever other's concepts or words are included, I have adequately cited and documented the original sources. I additionally declare that I have adhered to all principles of educational honesty and integrity and haven't misrepresented or made-up or falsified any idea/data/fact/source in my submission. I perceive that any violation of the above are going to be cause for disciplinary action by the Institute and may additionally evoke penal action from the sources that have thus not been correctly cited or from whom proper permission has not been taken when required.

Yash Baley

13D100034

Acknowledgement

I would like to express my sincere gratitude towards my guide, ***Prof. A. Subash Babu*** for his valuable suggestions, support, guidance and constant encouragement through words of motivation throughout the course of this project work. I would also like to thank my parents and my sister for their constant support. Also, I would like to thank my friends for giving ideas through discussions.

Abstract

Reliability of consumer durables is a very important factor in this competitive age. Consumer durable are expected to have a long useful life and hence the quality of the product is a major deciding factor in determining the success of product in the market. Often, Companies may offer a quality product, but failures are probable to happen hence they provide after-sales services until a certain period which is termed as warranty period. This helps in improving the customer satisfaction but there is increase in price of the product too. It is highly imperative to maintain a balance between warranty service and price of the product since they both determine customer satisfaction and profit margin for the manufacturer. Usage of statistical tools to assess this relationship has helped researchers from a long time. Recent advancement in data science and machine learning can be leveraged to improve the analysis. Failure prediction is also an important aspect for the company while looking from financial point of view. While aiming for an efficient data analysis, we need to make sure the quality of data is up to the mark. Data acquisition efforts in the consumer durables market are often low, resulting in incomplete data and data consisting of very low number of variables. Here, the effort is to gain a clear understanding of the previous work done in this field through a thorough literature review and come up with techniques to improve performance of the predictions done through these analyses by using more advanced techniques such as machine learning and data science.

List of Figures

List of Tables

Abbreviations

FRW	-	Free Replacement Warranty
PRW	-	Pro-Rata Warranty
1D/2D	-	One-Dimensional/Two-Dimensional
MTTF	-	Mean Time to Failure
MTBF	-	Mean Time between Failure
FMEA	-	Failure Modes and Effect Analysis
SUV	-	Sports Utility Vehicle
CPV	-	Customer Perceived Value
FSI	-	Failure Score Index
LOS	-	Level of Service

Content

Chapter 1

Introduction

1.0 Introduction

In this chapter, we'll introduce various concepts that are used in the report. Starting with the definition and basic mathematics behind the concepts. We'll also look into the current scenario of data analytics and machine learning. Aim and scope of the problem in hand will also be discussed towards the end of the chapter.

1.1 Warranty

While deciding to purchase a product, customers tend to compare the products offered by competing brands. In consumer durables market, products from different competitors have similar characteristics such as price and features. In such scenarios, customers focus more on post-sale factors such as warranty offered, part availability, servicing and maintenance cost. Out of these factors, warranty details are available to customer at the time of purchase.

In the case of a new product in market, uncertainty about the product performance exit and hence warranty helps in gaining the confidence of customers. These help in assuring useful life of product until the warranty period at least. As a result, it can be said that warranties play an important role in the commercial market by removing the hurdles of uncertainty. From the manufacturers perspective, warranties help in resolution of disputes. Manufacturers remain responsible for the failures that are decided at the time of purchase and the time/usage horizon is limited through this. Therefore, warranty is important from the manufacturer's as well as buyer's perspective.

1.1.1 Warranty Definition

A product warranty is an agreement between the seller and buyer, which establishes a liability between these parties in the event of failure. It specifies the expected performance of the product and the redress available to the buyer if a failure occurs. Here, the seller refers to the party responsible for assuring the warranty terms are met, and this is usually the manufacturer or retailer of the product. Then, the buyer is normally the ultimate paying consumer [1].

1.1.2 Warranty Policies- One Dimensional

Warranty policies consist of combinations of three variables i.e. mode of service, cost of service and dimensionality. Mode of service defines either the product will undergo repair or will be replaced with a new one, cost of service defines the cost borne by customer and manufacturer, and dimensionality means the number of dimensions along which the warranty is dependent. Along with these, warranty may be renewable or non-renewable depending on the contract. Here the warranty policies are one dimensional, i.e. they depend on one dimension only. Here the dimension taken for simplification is calendar time.

Numerous warranties are possible by combination of these variables. Some of the major warranty policies in the market are described below. [1]

Following notations are used in the following sections:

W = Warranty period

X = Time at failure

C = Cost of product

Policy 1

Non-Renewing free-replacement warranty: The seller repairs or provides replacements of product until time T (warranty period) from the time of purchase at zero cost to the customer. Warranty doesn't anew hence expire after time W. Let us say that the product fails at time X where X < W; hence the warranty is valid for period W-X. In case of additional failures, this process will be repeated until time T is reached. This warranty is famously abbreviated as FRW.

Policy 2

Basic Rebate Warranty policy: The seller agrees to refund αC amount to the customer, if the item fails before time W (warranty period) from the time of purchase. C is the cost of product and 0 < $\alpha < 1$.

Seller is not responsible for any repair or replacement in this case. The customer can use the cash refund to buy a replacement of the product. If α=1, this case will be equivalent to the "Money Back Guarantee" offer made by sellers.

Policy 3

Pro-Rata Renewing Warranty Policy: Pro-rata means proportional, hence the cost borne by the manufacturer is a function of the ratio of used life and warranty period. Let us say that the product has failed at time X where X < W; hence the cost paid by seller for a replacement item will be $(1 - X/W) * C$, if the item fails to achieve a lifetime of W. This policy is renewing since once a replacement is done, warranty is valid until the service time of the replacement is W. This warranty is famously abbreviated as PRW.

Policy 4

Combination FRW & PRW:

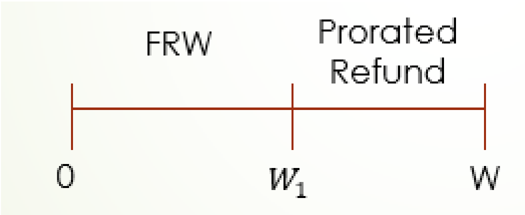


Figure 1.1 Combination FRW & PRW

This policy is a combination of the free replacement warranty (FRW) and Pro-rata warranty (PRW). Product is replaced/repaired free of cost if it fails before time W_1 and the seller provides a prorated refund to the customer if failure occurs in W_1 to W as shown in Fig. 1.1 The warranty doesn't renew in this case.

Policy 5

Non-Renewing three stage warranty:

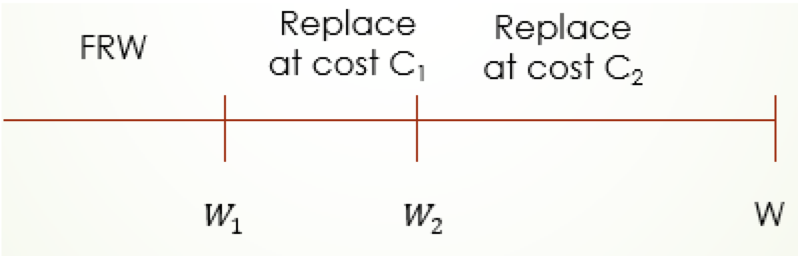


Figure 1.2 Non-Renewing three stage warranty

The seller provides repairs or replacement until time W_1 at zero cost, cost of repair/replacement is C_1 if failure occurs between time W_1 and W_2 and cost is C_2 if failure occurs between time W_2 to W from the time of purchase. The warranty policy is non-renewing. From Fig.1.3, the distribution of cost between along the age of the product/item can be clearly seen.

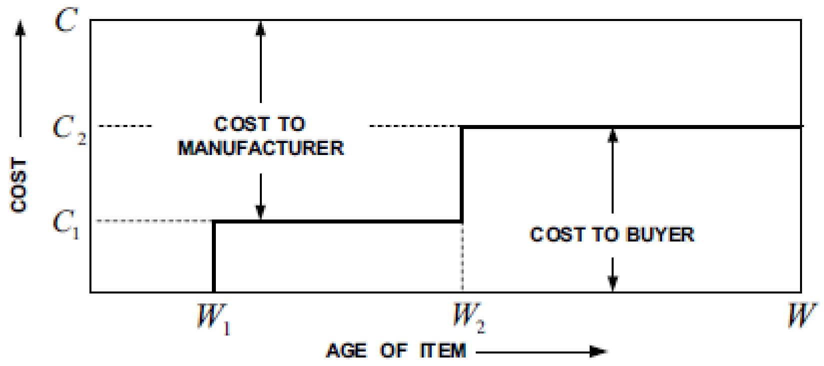


Figure 1.3 Repair/replacement costs under a three-stage warranty, Policy 5 [2]

Policy 6

Renewing FRW: This policy is similar to the Policy 1, i.e. manufacturer provides replacement/repair of the product if the product fails before warranty period, but here the warranty is renewed if the warranty claim is valid.

1.1.3 Warranty Policies- Two dimensional

In two dimensional warranties, the warranty period is defined using a 2-D plane. Usually one axis represents time and the other axis depicts the usage of the product. For example, when defining warranty for a vehicle, two dimensions are distance covered by the vehicle and the time from the date of purchase.

Policy 7

Two-Dimensional Non-Renewing FRW Policy: Seller makes an agreement to repair/replace products if failure occurs before time W or usage is less than U.

Fig. 1.4 illustrates this policy, Usage is along Y-axis and time period is along X-axis

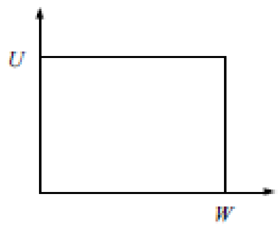


Figure 1.4 Two-Dimensional Warranty Policy – Usage versus time [2]

If the usage is heavy, the warranty can expire well before W, and if the usage is very light, then the warranty can expire well before the limit U is reached. Should a failure occur at age X with usage Y, it is covered by warranty only if X is less than W and Y is less than U. If the failed item is replaced by a new item, the replacement item is warrantied for a time period W – X and for usage U – Y.

Policy 8

Two-Dimensional Combination FRW/PRW: The seller agrees to provide replacements for failed items free of charge up to a time W1 from the time of the initial purchase provided the total usage at failure is below U1. Any failure, with time at failure greater than W1 but less than W2 and/or usage at failure greater than U1 but less than U2, is replaced at a prorated cost. Failures with time greater than W2 or usage greater than U2 are not covered by warranty.

Policy 9

Cumulative FRW: A lot of *n* items is warranted for a total (aggregate) period *nW*. The *n* items in the lot are used one at a time. If $S_n < nW$, free replacement items are supplied, also one at a time, until the first instant when the total lifetimes of all failed items plus the service time of the item then in use is at least *nW*.

This type of warranty is offered to products which are bought in lots, they are used one by one and others are kept as spares.

Policy 10

Cumulative FRW (More than one item at a time): This policy is similar to the Policy 9 but this policy is for a process which has more than one item in operation. Let us say there are k (less than n) items are used at a time out of the n items, and the remaining n-k items are kept as spares. Once a failure occurs, the failed item is replaced with a new one. Once all the spares are used, sellers agree to supply free replacements as necessary until total service time is greater than *nW*.

1.1 Data analytics

Data analytics brings speed and accuracy in any analysis. Now-a-days nearly all businesses are shifting their focus in the field of data analytics. Main factors that contribute in making data analytics a success are data availability and computational power. Talking about India, the steep growth in technological advancements and education level has catalyzed the progress of data analytics. Devices are becoming more interconnected and hence the amount of data being generated per unit time is increasing day by day. Same progress can be seen in the consumer durables market in terms of data richness. Hence, manufacturers need to utilize this maintain a competitive edge in the market. This data can be used to make predictions which can help manufacturers to make strategies that can improve customer satisfaction level, profit margin, reliability and market share along with many other parameters.

1.2 Machine learning

Machine learning is a technique based on algorithms which figure out patterns from data without being explicitly programmed. It is difficult to make predictions in consumer durables market since number of factors governing the system are very high and it is difficult to explicitly figure out effect of each factor to give a good prediction. This is where, machine learning can be leveraged, since these techniques takes care of the hurdles and can give out more efficient results with a relatively smarter approach.

1.3 Cash flow analysis

Cash flow statement gives information about where the money is spent (cash outflow) and from where the money is coming (cash inflow). In consumer durables market, manufactures have different channels for flow of cash. For example, cash inflow can be from EMIs, servicing cost, extender warranty purchases and cash outflow can be seen when repair/replace is being made for a warranty claim. Cash flow analysis is of vital importance to a manufacturing company since it can figure out if sales aren’t generating enough cash to pay for the expenses.

1.4 Aim and scope of the problem on hand

Data analytics possess great potential to tackle problems which are related to manufacturing. There is limited research in the field of warranty analytics where solutions are generated using data analytics. The primary reason for this being the non-availability of data and low-adaptability of technology. The recent developments in industry such as technology adaption in the consumer durables market and greater consumer-seller interaction has generated more data. All these things are proving to be helpful in the application of data analytics in this field.

For the same purpose, initial efforts were focused on the relevant literature review. To imitate the real situation, a system was developed to simulate the data and to carry out analysis to obtain insights. The objective of this exercise is to improve the customer satisfaction which is referred to as Customer Perceived Value while keeping the costs of operation low.

1.5 Outline of the report

This report presents a method to analyze the warranty policies in the consumer durables market. Initially the concept of warranty, machine learning and cash flow analysis is presented. Literature review of work carried out in the field of warranty so far is presented. Later problem in hand and approach is discussed. Later, progress that have been achieved till now is presented along with future work.

The chapter wise outline is as follows:

Chapter 2: Aimed at literature review of research carried out in the field of warranty analytics. Provides various techniques that have been applied by various researchers along with their results.

Chapter 3: The chapter puts forward the problem on hand that we aim to tackle along with the approach to solve the problem.

Chapter 4: In this chapter, various customer profiles were generated which will emulate the customer profiles in real situation. Based on these profiles, method to generated data insights was also shown.

Chapter 5: Identification of various failure modes and simulation of failure. Based on this data generated, method to derive insights and predictions was shown.

Chapter 6: Definition of Customer Perceived Value along with various factors affecting it. Method to derive insights from the customer ratings was also shown.

Chapter 7: Identifying costs for each type of failure mode and generating cash flow for Customer, Service provider and Manufacturer.

Chapter 8: Algorithm used for optimizing CPV with costs was explained, using framework and process flow and results of the algorithm were shown.

Chapter 9: Summarizing the work done and proposed future scope of work.

Chapter 2

Literature Review

2.0 Introduction

To make advancements in the field of warranty management using data analytics and machine learning, it is imperative to learn the basic concepts behind these field. In this chapter, we'll briefly define the concepts and terminologies learnt from various books, research paper and online sources.

2.1 Reliability

This is the probability that an item will perform its indicated mission without failing for the expressed time when utilized as indicated by the specified conditions. Following sections cover the relevant topics that are useful for our study.

2.1.1 Bathtub Hazard Rate Curve

Bathtub hazard rate curve is an outstanding idea to speak to failure rates of different engineering product in light of the fact that the failure rate of these things changes with time. Its name come from its shape being similar to a bathtub as appeared in Fig. 2.1. Three different sections of the curve are observed in the figure: degradation region, useful-life region, and wear out region. These areas indicate three stages that a new product goes through amid its life time.

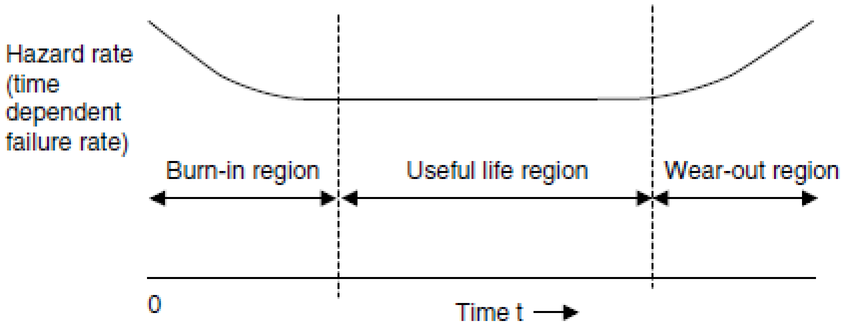


Figure 2.1 Bathtub hazard rate curve [2]

The general hazard rate is given by the following equation

$$
\begin{aligned}
\lambda(t) &= \frac{f(t)}{R(t)} \\
&= \frac{f(t)}{1-F(t)} \\
&= \frac{f(t)}{1-\int_0^t f(t)\,dt}
\end{aligned}
\tag{2.1}
$$

where

$\lambda(t)$ = item hazard rate (i.e., time t dependent failure rate)

$f(t)$ = item failure density function (probability density function)

$F(t)$ = cumulative distribution function (i.e., the item failure probability at time t)

$R(t)$ = item reliability at time t

Table 2.1 compiles information about hazard rate and reliability function for Exponential distribution, Weibull distribution and a general distribution:

Table 2.1 Hazard Rate Functions and Probability Functions [5]

Distribution	PDF	Hazard Rate	Reliability
Exponential	$f(t)=\lambda e^{-\lambda t}, \quad t\geq 0, \lambda>0$	$\lambda(t)=\frac{\lambda e^{-\lambda t}}{1-\int_0^t \lambda e^{-\lambda t}dt} = \lambda$	$R(t)=e^{-\int_0^t \lambda\, dt} = e^{-\lambda t}$
Weibull	$f(t)=\frac{\theta t^{\theta-1}}{\alpha^\theta}e^{-(t/\alpha)^\theta}, \quad t\geq 0, \alpha>0, \theta>0$	$\frac{\theta}{\alpha^\theta}t^{\theta-1}$	$R(t)=e^{-\int_0^t \frac{\theta}{\alpha^\theta}t^{\theta-1}dt} = e^{-\left(\frac{t}{\alpha}\right)^\theta}$
General	$f(t)=[c\lambda\gamma t^{\gamma-1}+(1-c)\theta t^{\theta-1}\mu e^{\mu t^\theta}][\exp[-c\lambda t^\gamma-(1-c)(e^{\mu t^\theta}-1)]]$ for $0\leq c\leq 1$ and $\gamma,\theta,\mu,\lambda>0$	$\lambda(t)=c\lambda\gamma t^{\gamma-1}+(1-c)\theta t^{\theta-1}\mu e^{\mu t^\theta}$	$R(t)=e^{-\int_0^t\left[c\lambda\gamma t^{\gamma-1}+(1-c)\theta t^{\theta-1}\mu e^{\mu t^\theta}\right]dt}$ $R(t)=\exp[-c\lambda t^\gamma-(1-c)(e^{\mu t^\theta}-1)]$

2.1.2 Mean Time to Failure

Mean time to failure is a predicted measure which tell us the expected time to failure of a system. Mean time between failure is the average of times to failure. Mathematically it is given by

$$\text{MTTF} = \int_0^\infty t\,f(t)\,dt \tag{2.2}$$

or

$$\text{MTTF} = \int_0^\infty R(t)\,dt \tag{2.3}$$

where

MTTF = Mean time to failure

2.1.3 Failure Data Collection

Failure data can provide important insights and form the backbone of analysis related to reliability. There are various sources for collecting data. Following table summarizes different ways of collecting data which will help in reliability analysis:

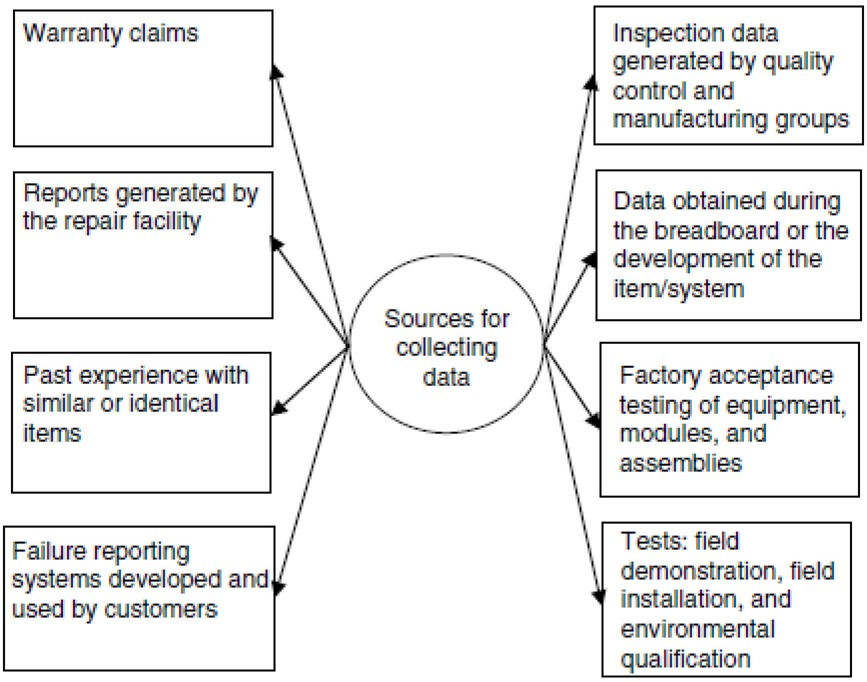


Figure 2.2 Sources for Collecting Data [2]

2.1.4 Failure Mode and Effect Analysis (FMEA)

To measure reliability of engineering systems, failure mode and effect analysis is extensively used. Failure flow chart outlines a step-by-step process to perform this analysis

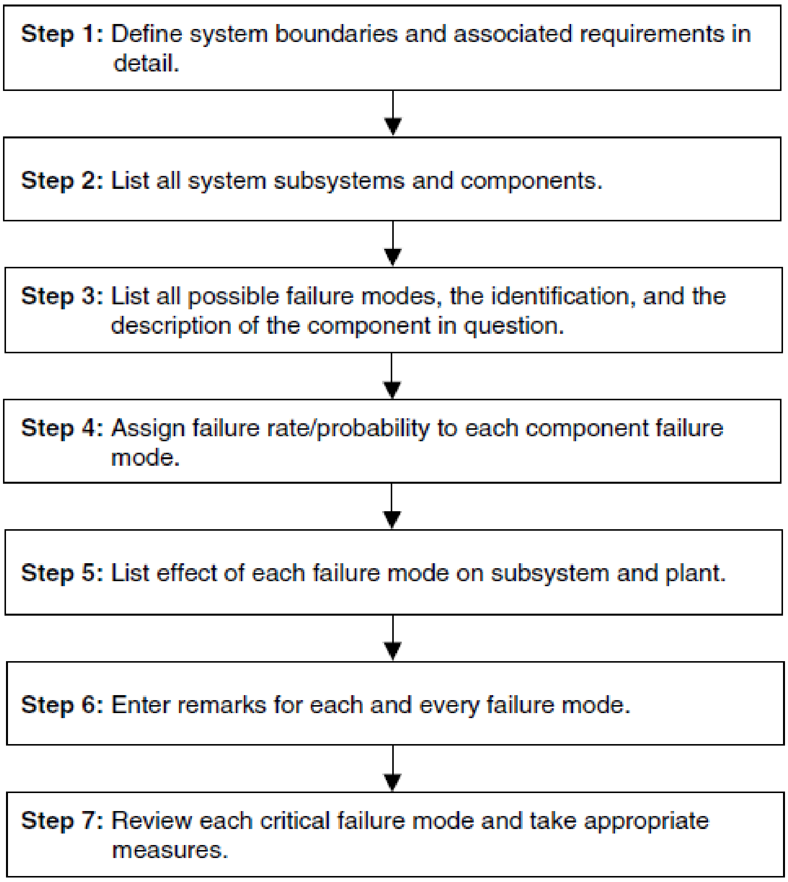


Figure 2.3 Steps for performing FMEA[2]

2.2 Analysis of Warranty Claim Data

Numerous research papers have been published in this field. Hence it is beneficial to bucket the papers into different sections depending on their approach of tackling the problem. Following are the sections:

2.2.1 Analysis Based on Age of Product

There can be many factors that play an important role in deciding the warranty policy. Age being the most commonly and hence it is imperative research about it. Robinson and McDonald, 1991 [3]; Kalbfleisch and Lawless, 1996 [4]; Lawless, 1998 [5]; Karim et al., 2001a, b [6]; Kalbfleisch et al., 1991 [7] and many other papers are focused in this area.

Kalbfleisch et al., 1991 [7] has discussed the method where we assume the process to be Poisson, and try to fit a Poisson model using the data available. It has incorporated a concept of "reporting lag" which is basically the time difference between time when the product failed and time when the warranty was claimed for the product. Here the products are taken as cars

Following variables have been used:

N_x = Number of cars entered the service on day x

n_{xtl} = Number of claims at age t which entered service on day x with a reporting lag of l

Then the distribution of n_{xtl} is poisson in nature i.e.

$$n_{xtl} \sim \text{Poisson}(\mu_{\text{xtl}})$$

where,

Also,

$$\mu_{\text{xtl}} = N_x.\lambda_t.f_l \tag{2.4}$$

Constraint,

$$x + t + l < T \tag{2.5}$$

where

μ_{xtl} = Mean of the Poisson distribution

λ_t is the expected number of claims for a car at age t

f_l is the probability that the repair claim enters the database used for analysis l days after it takes place

T is the current date

We can then write the likelihood

$$L = \prod\prod\prod_{x+t+l \leq T} - \frac{e^{-N_x \lambda_t f_l}(N_x \lambda_t f_l)^{n_{xtl}}}{n_{xtl}!}. \tag{2.6}$$

To estimate λ_t, Lawless & Kalbfleisch, 1992 [4] has suggested the following formula

$$\hat{\lambda}_t = \frac{\sum_{x=0}^{T-t} n_{xt}}{\sum_{x=0}^{T-t} N_x}, \quad t = 0, 1, \ldots, \tag{2.7}$$

Lawless, 1998 [5] & Kalbfleisch and Lawless, 1996 [4] has estimated λ_t as following

$$\hat{\lambda}(a) = \frac{n^T(a)}{R^T(a)}, \quad a = 0, 1, \ldots, \tag{2.8}$$

where,

$$n^T(a) = \sum_{d=0}^{T-a} n^T(d, a) \tag{2.9}$$

is the number of claims at age a which are reported up to T number of days, and $n^T(d,a)$ is total number of claims which are reported at age a for units which were sold on day d

2.2.2 Analysis Based on Aggregate Information about Warranty Claims

In consumer durables market, the data available is generally in aggregate forms. Hence we need to focus more on techniques which provide analytical solution to aggregate form of data. Trindade & Haugh, 1980 [8] have used a renewal process which estimates the reliability of components with an assumption that once a component fails it is replaced by a new one immediately.

$$M(t) = F(t) + \int_0^t M(t-x)\,dF(x), \tag{2.10}$$

Or,

$$F(t) = M(t) + \int_0^t M(x)\, \mathrm{d}F(t-x). \tag{2.11}$$

where,

F(t) = Cumulative probability distribution for a component having lifetime of t

M(t) = Component renewal function (or expected number of replacements during time t)

In this method, *F(t)* is estimated using estimate of *M(t),* and to do that numerical deconvolution techniques can be used.

Sometimes, aggregate information is available but

2.2.3 Analysis of Two-Dimensional Warranty

As explained earlier in Chapter 1, Two-Dimensional warranty or 2-D warranty incorporates age as well as usage generally. Most common example being the vehicles. For example, A Maruti Suzuki hatchback has a warranty of 2 years and 40,000 km (whichever comes first).

Moskowitz & Chan, 1994 [9] has given a method which employ Poisson regression model.

$$\Pr[n_i] = \frac{\mu_i^{n_i} e^{-\mu_i}}{n_i!}, \tag{2.11}$$

where,

$\Pr[n_i]$ = Probability of event happening n_i number of times

$\mu_i = f(\boldsymbol{X_i}, \beta)$ with i = 1,2,3…,m and n_i is regression function of the age and usage amount,

β is coefficient vector of regression model.

Moskowitz and Chan have also suggested the following regression models

X_{i1} is age, X_{i2} is mileage

Multiple linear form-

$$\mu_i = \beta_1 X_{1i} + \beta_2 X_{2i} \tag{2.12}$$

Log-linear form-

$$\mu_i = \exp(\beta_1 X_{1i} + \beta_2 X_{2i}); \tag{2.13}$$

Power-linear form-

$$\mu_i = \beta_0 X_{1i}^{\beta_1} + X_{2i}^{\beta_2}. \tag{2.14}$$

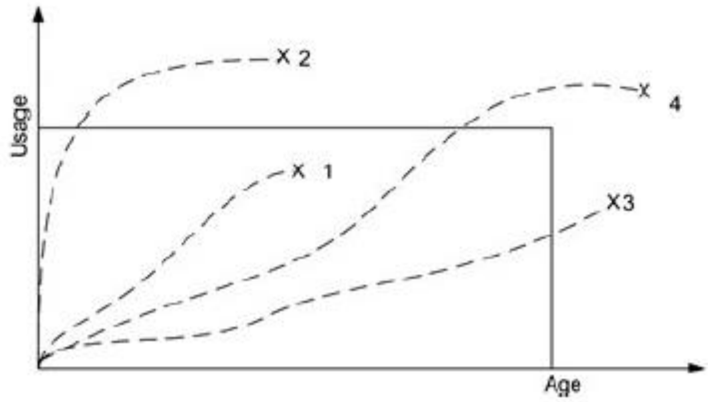


Figure 2.1 Two-Dimensional Warranty

Fig.2.4 illustrates various possible situations in 2-D warranty. Warranty is valid until is it in the defined rectangle. We can see that X_2, X_3, X_4 are out of warranty irrespective of their difference in usage rates.

A. Kleyner and P. Sandborn [10] have kept usage time as their primary variable and mileage accumulation is calculated using the data from the claims made. Mathematically, if we multiply the cumulative density function (cdf) of time based warranty model with mileage based warranty model we can get a 2-d warranty model.

$$F(t)_{warranty} = F(t)_{time\text{-}based} * F(t)_{mileage\text{-}based} \tag{2.15}$$

$F(t)_{time\text{-}based}$ is cumulative density function, this function is derived in Kleyner & Sandborn, 2005 and is given as

$$F(t)_{time\text{-}based} = 1 - e^{-\left(1+\frac{\beta(t-t_S)}{t_S}\right)\left(\frac{t_S}{\eta}\right)^{\beta}} \tag{2.16}$$

where $t \geq t_S$

t_S = Hazard rate stabilization point

β = Weibull slope of the failures observed before the time t_S

η = Weibull scale parameter of the failures observed before the time t_S.

As shown in Fig.2.5 the cumulative probability density function shows an increase in the probability that vehicle will reach mileage limit M_o with increase in time.

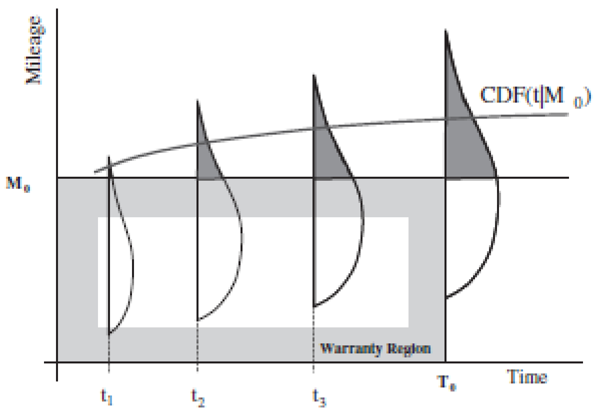


Figure 2.2 Vehicle Mileage accumulation in 2-D warranty [10]

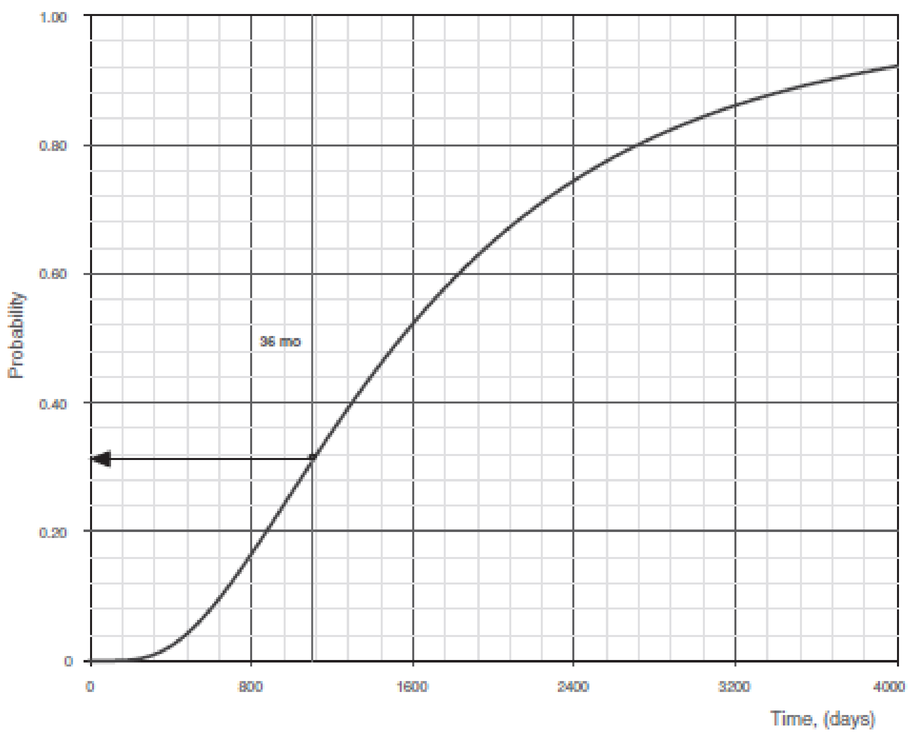


Figure 2.3 Probability of exceeding 36000 miles (based on automotive dealership data) [10]

Kleyner and Sandborn compiled data from automotive dealer. They found the distribution of probability as show in Fig.2.4 below. They also found out that the number of data points required for a legit estimation of parameters depends on the sufficient convergence. They concluded the paper by suggesting that most of the dealership records are from the products that have failed at some time, hence the mileage of the product might be affected due to the failure. Therefore, analysts need to make sure that they don't include only single type of failure since it can cause bias in the result and hence opt for data from multiple failure modes.

In one of the working paper by Majeske, 2003 [11]. They have used Non-Homogenous Poisson Process (NHPP) to analyze the warranty claims. In NHPP, intensity function *v(t)* is defined and it is as follows:

$$v(t) = (\alpha t)^{\beta} \tag{2.17}$$

Also, Crow in 1974 [12] showed that the first time to failure can be approximated by a Weibull distribution.

$$F(t) = 1 - e^{-(\alpha t)^{\beta}} \tag{2.18}$$

where, *F(t)* is the failure probability.

Majeske observed results as shown in Fig.2.5 for the hazard rate.

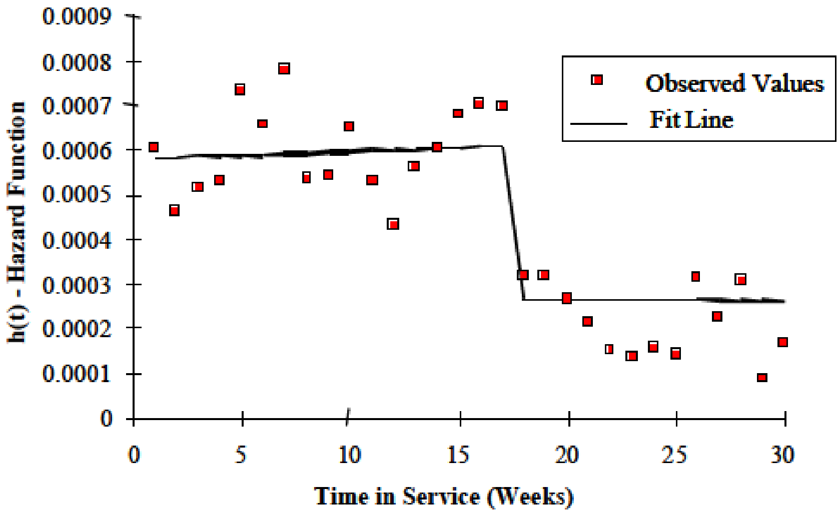


Figure 2.4 A typical hazard rate

2.3 Analysis of Warranty Costs

Kim and Rao, 2000 [13] aimed to find the expected warranty cost of 2-D free-replacement warranty using a bivariate exponential distribution. The warranty policy is same as Policy 1 described in section 1.1.3. The derivation of function is too mathematically involved. Vickie Lee Hill et al., 1991 [14] gave a simulation model for analyzing the warranty. They have simulated warranty by assuming lifetime distribution such as Weibull, normal, gamma etc. They have devised a step by step approach for finding expected costs of warranty by assigning probability to each possibility. Also, they have incorporated the concept of random numbers for simulation which helps in making the simulation.

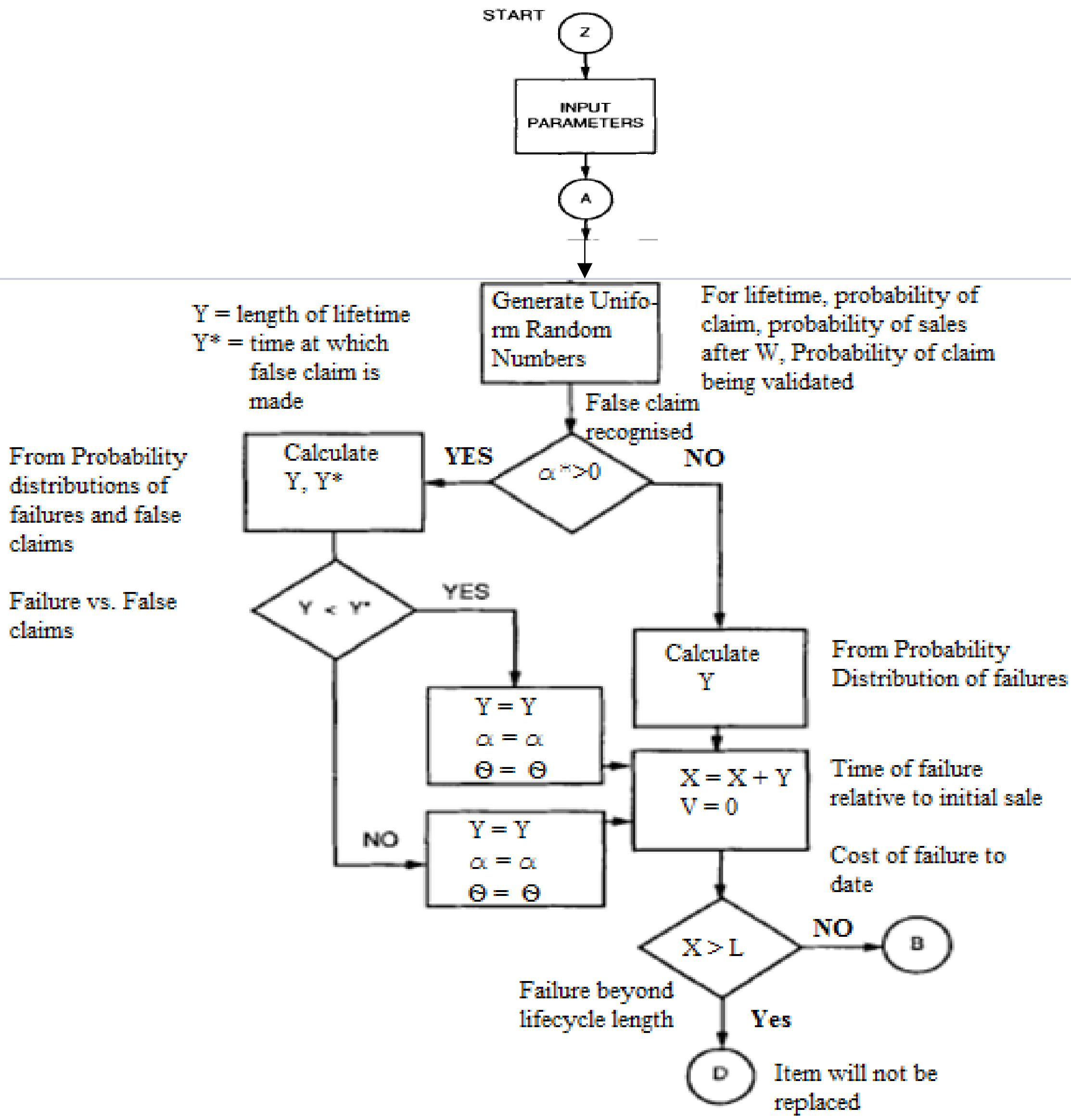


Figure 2.5 Simulation model for cost analysis

Zhiwei Chen et al., 2017 [15] have proposed a comprehensive warranty cost model that considers burn-in, FRW and PRW as its 3 phases and failure occurs in 2 types i.e. minimal and catastrophic. Fig.2.6 shows framework of the model. First the product undergoes a burn-in testing. If the product doesn't fail during this phase it is sent to the seller. Otherwise there are two types of failure possible namely type I and type II. Once the product is sold it has the same warranty policy as Policy 4 in section 1.1.2.

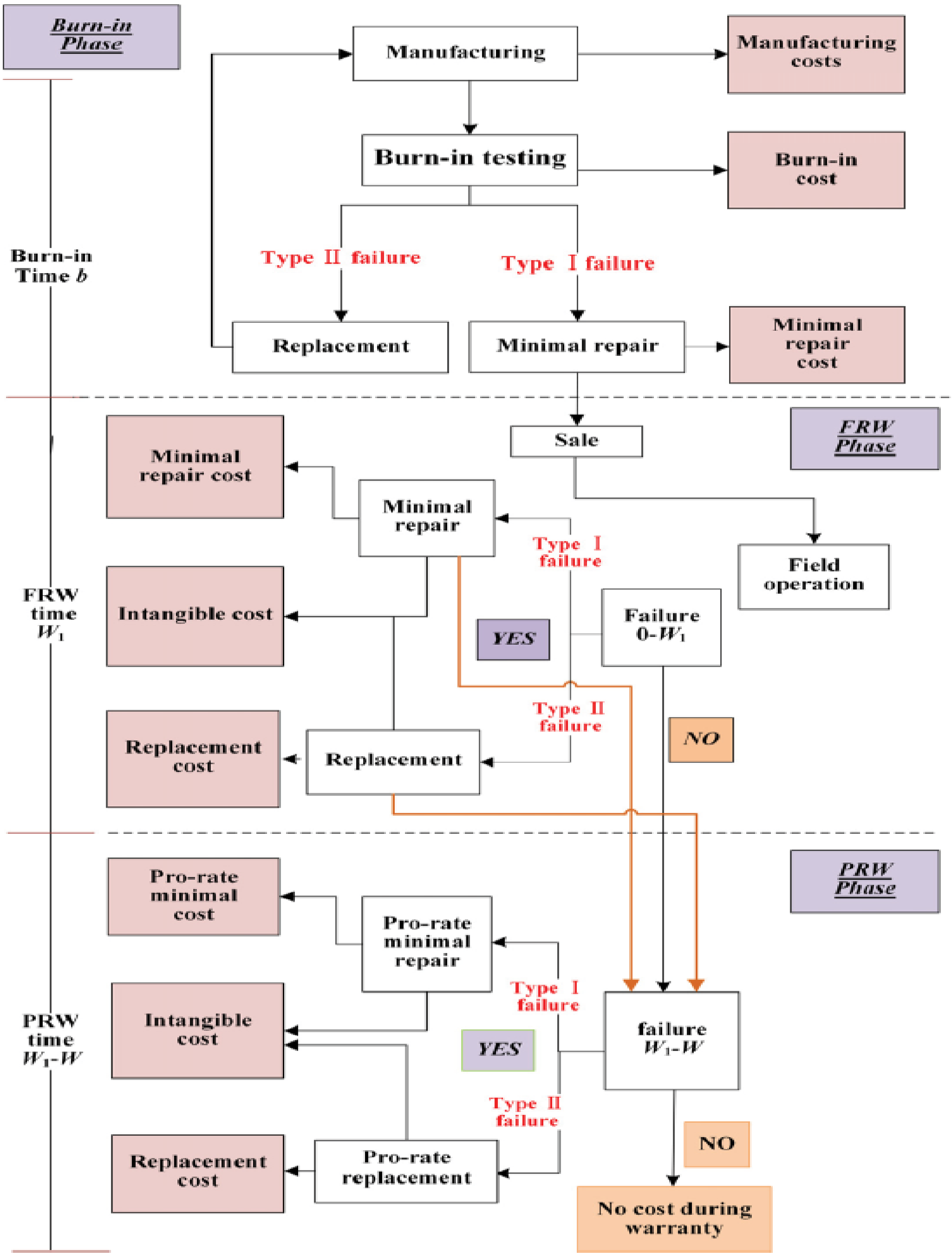


Figure 2.6 Framework for warranty cost analysis [15]

2.4 Machine learning and Data Analytics

As already stated, the main aim of the report is to apply machine learning techniques to solve the problem of warranty analytics and one of the major reason behind this approach is increasing availability of data. Fig.2.10 shows that data is growing at a 40 percent compound annual rate reaching nearly 45 ZB by 2020. This section is aimed at literature review about the concepts of machine learning and the advantages of machine learning over the statistical analysis.

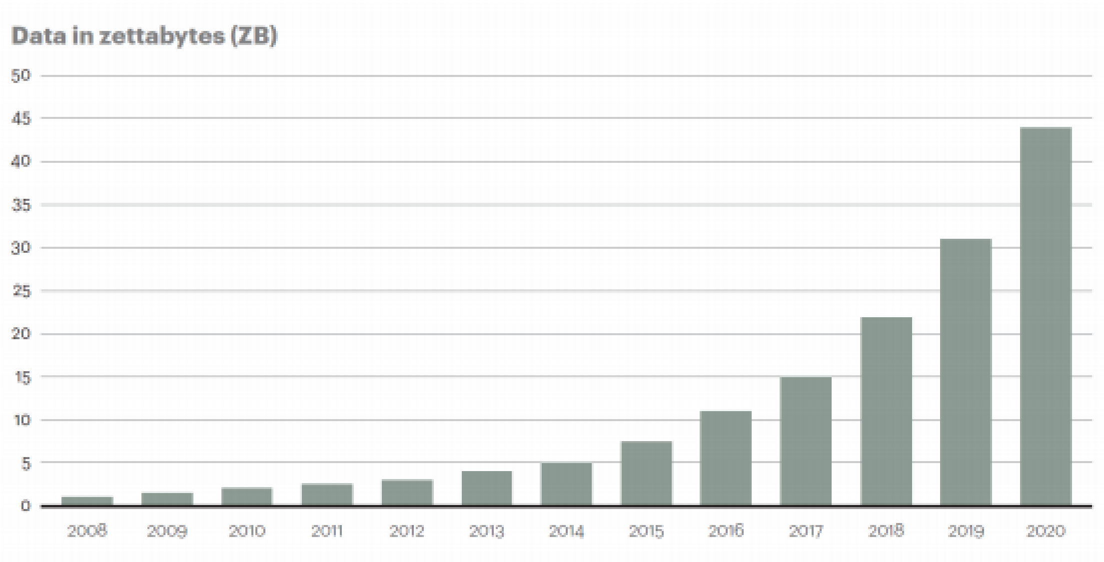


Figure 2.10 Increase in volume of data with time [18]

Two definitions of Machine Learning are offered. Samuel [16] described it as: "the field of study that gives computers the ability to learn without being explicitly programmed." This is an older, informal definition.

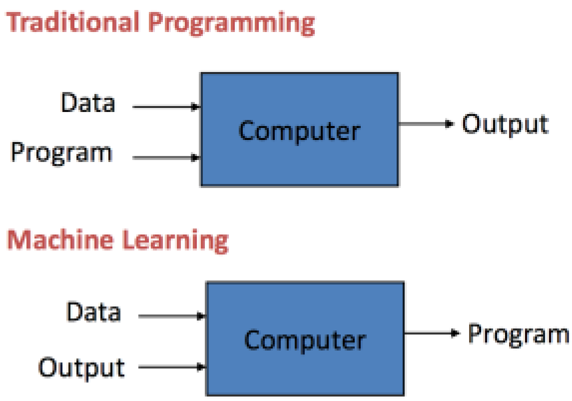


Figure 2.11 Traditional Programming vs. Machine Learning

Mitchell [16] provides a more modern definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Example: playing checkers.

E = the experience of playing many games of checkers

T = the task of playing checkers.

P = the probability that the program will win the next game.

In general, any machine learning problem can be assigned to one of two broad classifications: Supervised learning and Unsupervised learning.

In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output. Supervised learning problems are categorized into "regression" and "classification" problems. In a regression problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function. In a classification problem, we are instead trying to predict results in a discrete output. In other words, we are trying to map input variables into discrete categories.

Unsupervised learning allows us to approach problems with little or no idea what our results should look like. We can derive structure from data where we don't necessarily know the effect of the variables. We can derive this structure by clustering the data based on relationships among the variables in the data. With unsupervised learning there is no feedback based on the prediction results.

Every machine learning algorithm has three components:

- Representation: how to represent knowledge. Examples include decision trees, sets of rules, instances, graphical models, neural networks, support vector machines, model ensembles and others.
- Evaluation: the way to evaluate candidate programs (hypotheses). Examples include accuracy, prediction and recall, squared error, likelihood, posterior probability, cost, margin, entropy k-L divergence and others.

- Optimization: the way candidate programs are generated known as the search process. For example, combinatorial optimization, convex optimization, constrained optimization.

2.4.1 Methods in Machine Learning

Following are the major methods used in machine learning-

1. Linear Regression

It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation

$$Y = a * X + b \tag{2.19}$$

where,

Y - Dependent Variable

a - Slope

X - Independent variable

b - Intercept

2. Logistic Regression

It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s). In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function. Hence, it is also known as logit regression. Since, it predicts the probability, its output values lie between 0 and 1 (as expected).

Here is an example of logistic regression, the independent variable is age and g is known as a link function.

$$g(y) = \beta o + \beta(Age) \tag{2.20}$$

Since probability must always be a non-negative number, put the linear equation in exponential form. For any value of slope and dependent variable, exponent of this equation will always be positive or zero.

$$p = \mathrm{e}^{(\beta o + \beta(Age))} \tag{2.21}$$

To covert p into probability, insert a denominator with the value as shown

$$p \;= \frac{e^{(\beta o\,+\,\beta(Age))}}{e^{\beta o\,+\,\beta(Age)}+1} \tag{2.22}$$

Replacing the linear equation with its equivalent, that is y

$$p \;= \frac{e^{y}}{1+e^{y}} \tag{2.23}$$

If p is probability of success and hence q = 1 – p is probability of failure

$$q \;=\; 1 \;-\; p \;=\; 1 \;-\{\frac{e^{y}}{1+e^{y}}\} \tag{2.24}$$

$$\frac{p}{1-p} = \; e^{y} \tag{2.25}$$

Convert the equation to logarithmic form as shown, now here

$$\log\left(\frac{p}{1-p}\right) = \; y \tag{2.26}$$

Where,

y – result through linear regression

p – probability of success

A typical logistic model is shown as below in figure

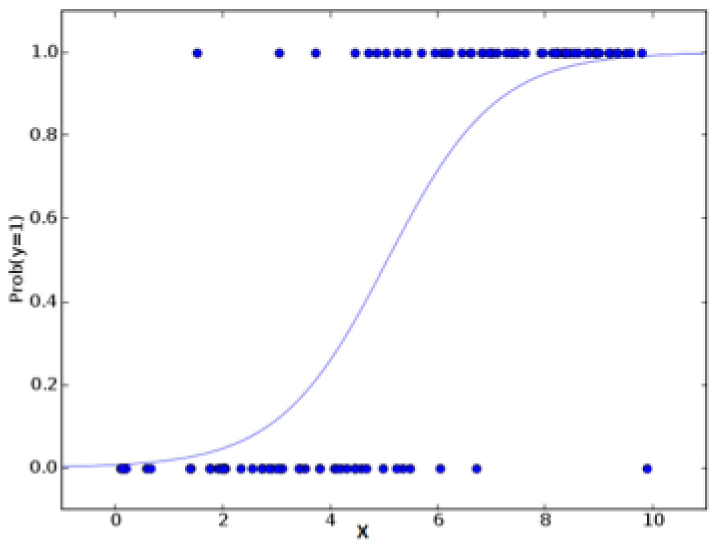


Figure. 2.12 Typical logistic model

3. Support Vector Machine

Support Vector Machine (SVM) can be used for both classification and regression challenges. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a coordinate. Now, we will find some hyper-plane that splits the data between the two differently classified groups of data. This will be the plane such that the –

A. It segregates the two classes better

B. Distances of the closest point in each of the two groups will be farthest away

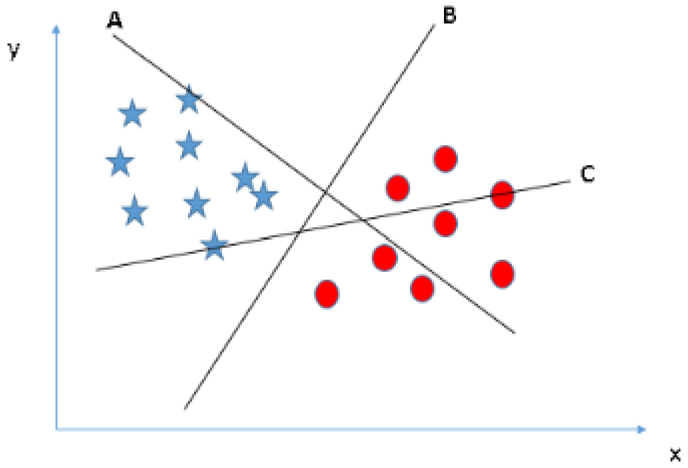


Figure. 2.13 Plane B segregates the classes better

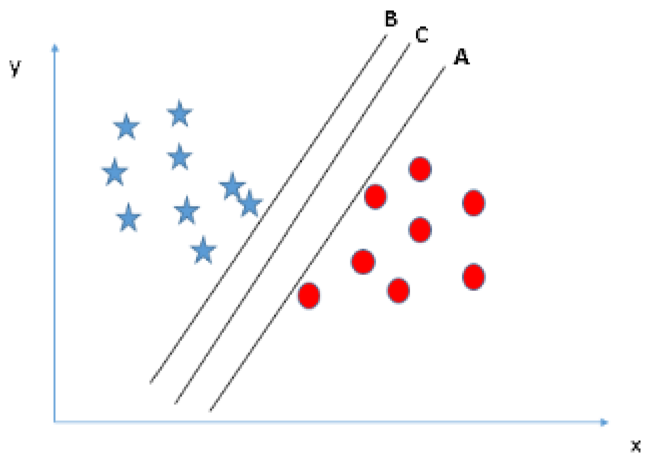


Figure 2.14 Plane C has the largest distance for the nearest points

Meinzer et. al, 2017 [17] has applied machine learning methods to predict the consumer satisfaction level. They setup a machine learning problem that compared 5 classifiers and analyzed data from 19,008 real service visits from an automotive company. The 105 extracted features were drawn from the most significant available sources: warranty, diagnostic, dealer system and general vehicle data. The best result for customer dissatisfaction classification was 88.8% achieved with the SVM classifier (RBF kernel). Furthermore, the 46 most potential indicators for dissatisfaction were identified by the evolutionary feature selection. The authors investigated different techniques to predict customer churn and concluded that support vector machines (SVM) showed the highest accuracy.

4. KNN

It can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. The case being assigned to the class is most common amongst its K nearest neighbors measured by a distance function. These distance functions can be Euclidean, Manhattan, Minkowski and Hamming distance. First three functions are used for continuous function and fourth one (Hamming) for categorical variables. If K = 1, then the case is simply assigned to the class of its nearest neighbor. At times, choosing K turns out to be a challenge while performing KNN modeling.

2.4.2 Confusion matrix

Confusion matrix defines the results obtained from classification algorithms. A typical confusion matrix for a binary classifier is shown in table 2.3-4.

Table 2.2 Example of Confusion matrix

		Predicted	
		Positive	Negative
Actual	Positive	210	12
	Negative	9	18

As can be seen from the table, it summarizes the classification results in a tabular format for better understanding. This table also helps in calculating various terms such as true positives, true negative, false positive, false negative, accuracy, sensitivity and specificity. The descriptions of them are given below-

a. True positive – When the prediction is positive and actual is positive too
b. True negative – When the prediction is negative and actual is negative too
c. False positive – When the prediction is positive but the actual is negative
d. False negative – When the prediction is negative but the actual is positive
e. Accuracy – Index measuring the accuracy of the classifier

$$Accuracy \ = \frac{True\ Positive+True\ Negative}{Total} \tag{2.27}$$

f. Sensitivity – Index measuring true positive rate

$$Sensitivity = \frac{True\ Positive}{Actual\ Positive} = \frac{True\ Positive}{True\ Positive+False\ Negative} \tag{2.28}$$

g. Specificity – Index measuring true negative rate

$$Specificity = \frac{True\ Negative}{Actual\ Negative} = \frac{True\ Negative}{True\ Negative+False\ Positive} \tag{2.29}$$

2.4.3 ShinyR

Shiny is an R package which helps in making interactive webpages. There are two components in Shiny namely UI and Server. UI is the user interface, it has many features including options to upload data, plot graphs, interactive buttons etc. The other component of Shiny is the Server which takes care of all the computations. Automatic "reactive" binding between inputs and outputs and extensive prebuilt widgets make it possible to build beautiful, responsive, and powerful applications.

RStudio software was used to implement R codes and Shiny. Few of the basic commands used for Shiny are given below

I. To install Shiny package in R

```
install.packages('shiny')
```

II. Basic Shiny app format

```
library(shiny) ## Loading the Shiny package, and other packages which are
               ##used
```

```
ui <- fluidPage (
## Define actions, buttons, plots to be implemented on the user interface
here
)
server <- function(input, output, session){
## Define working of all required actions
}
shinyApp(ui = ui, server = server)
```

III. Example of a Shiny app

UI part of the app

```
# Define UI for app that draws a histogram ----
ui <- fluidPage(

  # App title ----
  titlePanel("Hello Shiny!"),

  # Sidebar layout with input and output definitions ----
  sidebarLayout(

    # Sidebar panel for inputs ----
    sidebarPanel(

      # Input: Slider for the number of bins ----
      sliderInput(inputId = "bins",
                  label = "Number of bins:",
                  min = 1,
                  max = 50,
                  value = 30)

    ),

    # Main panel for displaying outputs ----
    mainPanel(

      # Output: Histogram ----
      plotOutput(outputId = "distPlot")

    )
  )

)
```

Server part of the app

```
# Define server logic required to draw a histogram ----
server <- function(input, output) {

  # Histogram of the Old Faithful Geyser Data ----
  # with requested number of bins
  # This expression that generates a histogram is wrapped in a call
  # to renderPlot to indicate that:
  #
  # 1. It is "reactive" and therefore should be automatically
  #    re-executed when inputs (input$bins) change
  # 2. Its output type is a plot
  output$distPlot <- renderPlot({

    x    <- faithful$waiting
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    hist(x, breaks = bins, col = "#75AADB", border = "white",
         xlab = "Waiting time to next eruption (in mins)",
         main = "Histogram of waiting times")

    })


}
```

Running the app

```
shinyApp(ui, server)
```

The above command lines can be saved in a file named app.R which will enable us to run the app directly by clicking the “Run App” button on the RStudio software. The result of running the app is shown below in figure 2.16.

2.4.4 Other packages used in R

Apart from shiny, a large number of other useful R packages. These packages are listed below

Table 2.3 Important libraries with description

Library name	Description
readxl	Import excel files into R
shinyjs	Perform common useful JavaScript operations in Shiny apps
shinythemes	Themes for use with Shiny. Includes several Bootstrap themes
ggplot2	A system for 'declaratively' creating graphics,
shinydashboard	Create dashboards with 'Shiny'. This package provides a theme on top of 'Shiny', making it easy to create attractive dashboards.
corrplot	A graphical display of a correlation matrix or general matrix

Table 2.3 continued

tableHTML	A tool to create and style HTML tables with CSS
DT	Data objects in R can be rendered as HTML tables using the JavaScript library 'DataTables' (typically via R Markdown or Shiny)
randomForest	Classification and regression based on a forest of trees using random inputs
dplyr	A fast, consistent tool for working with data frame like objects, both in memory and out of memory.
caret	Misc functions for training and plotting classification and regression models.
e1071	Functions for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines,
lattice	A powerful and elegant high-level data visualization system inspired by Trellis graphics, with an emphasis on multivariate data
rintrojs	A wrapper for the 'Intro.js' library
ggthemes	Some extra themes, geoms, and scales for 'ggplot2'
tidyverse	The 'tidyverse' is a set of packages that work in harmony because they share common data representations and 'API' design
cluster	Methods for Cluster analysis
factoextra	Provides some easy-to-use functions to extract and visualize the output of multivariate data analyses
ggfortify	Unified plotting tools for statistics commonly used, such as GLM, time series

2.5 Conclusion

There is substantial amount of research in the field of warranty analysis using statistical methods. This has helped in discovering various aspects which govern the warranty costs in the consumer durables market. Research focused on warranty analysis are very less in number and there is a high need to develop methods in the same to check the performance of machine learning in these areas. It is studied to incorporate various types of cost by accounting all the possibilities in the warranty. Various frameworks which can help in formulating warranty analysis for a general case are also studied. Thorough reading of machine learning algorithms along with application in R language was also done.

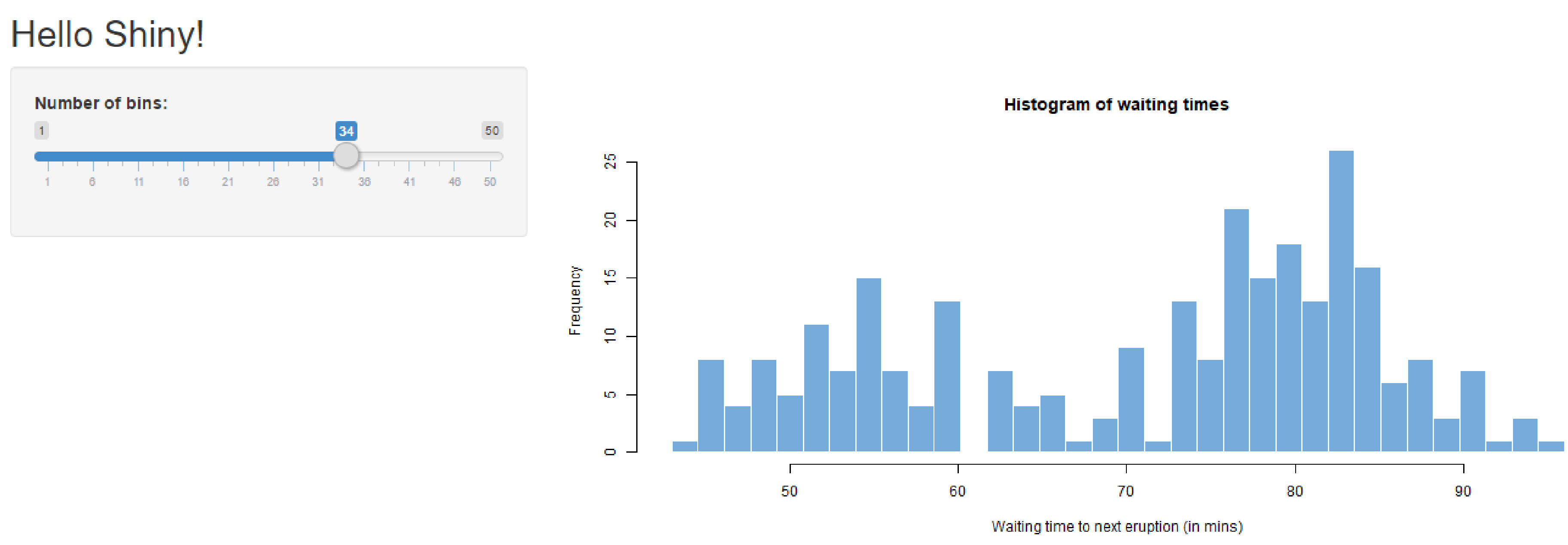


Figure 2.16 UI of a basic Shiny App

Chapter 3

Problem Statement & Approach

3.0 Introduction

The main problem statement is analyzing the available customer data and derive various insights from it. This will be done using modern techniques such as data analytics and machine learning algorithms. Cash flow analysis and prediction is also aimed which will help in management of the financial aspect of the manufacturing company. Keeping all these factors in mind, an index know as Customer Perceived value will be calculated, analyzed and various ways to improve the index shall be given.

3.1 Motivation

The market size of the consumer durables market is very huge, hence the amount of data that can be generated is also of various types and often voluminous. Consumer durables market covers almost the entire nation hence market size is very huge. as highlighted in the previous two chapters warranty is an important part of the consumer durables market therefore research and development in this field result in betterment of the manufacturer and the consumer. Chapter 2 of this report has highlighted the literature review in the field, but it was observed that there is not much work done where machine learning and data science is used. Industry today is becoming more and more inclined towards the Data Analytics therefore concentrated efforts are required in this field.

3.2 Approach

Since the research is carried out for the Indian consumer durables market therefore the initial records were focused at learning more about the Indian consumer durables market. For simplicity a major two-wheeler manufacturing company was taken as subject. To emulate real life situation, all the data necessary for analysis was generated using simulation. Various constraints were kept in mind while simulating the data such that the data generated shall imitate the real-life data as much as possible. A framework for analyzing the data and deriving various insights was made. In the end, an algorithm was suggested which was based on the all the insights generated from the simulated data. This algorithm tries to increase CPV while keeping the costs as much low as possible.

3.3 Process flow

The main aim of the report is to analyze and improve warranty service using data science and machine learning techniques. This kind of approach is not very known in the field of consumer durable analytics. These techniques have ability to handle large amount of data and derive efficient insights from it.

Data is the necessity while applying data analytics techniques. Since getting data from the market was not possible, hence the focus was shifted to simulation of data and then proceed with the analysis. The chapters 4, 5 and 6 encompasses steps taken while simulating the data and all the details related to data simulation including description of variables and parameters and the reasons behind choosing them.

Figure 3.1 schematically represents a process map describing the steps taken for the data simulation, the details of which are presented in Chapter 4, 5, 6 covers

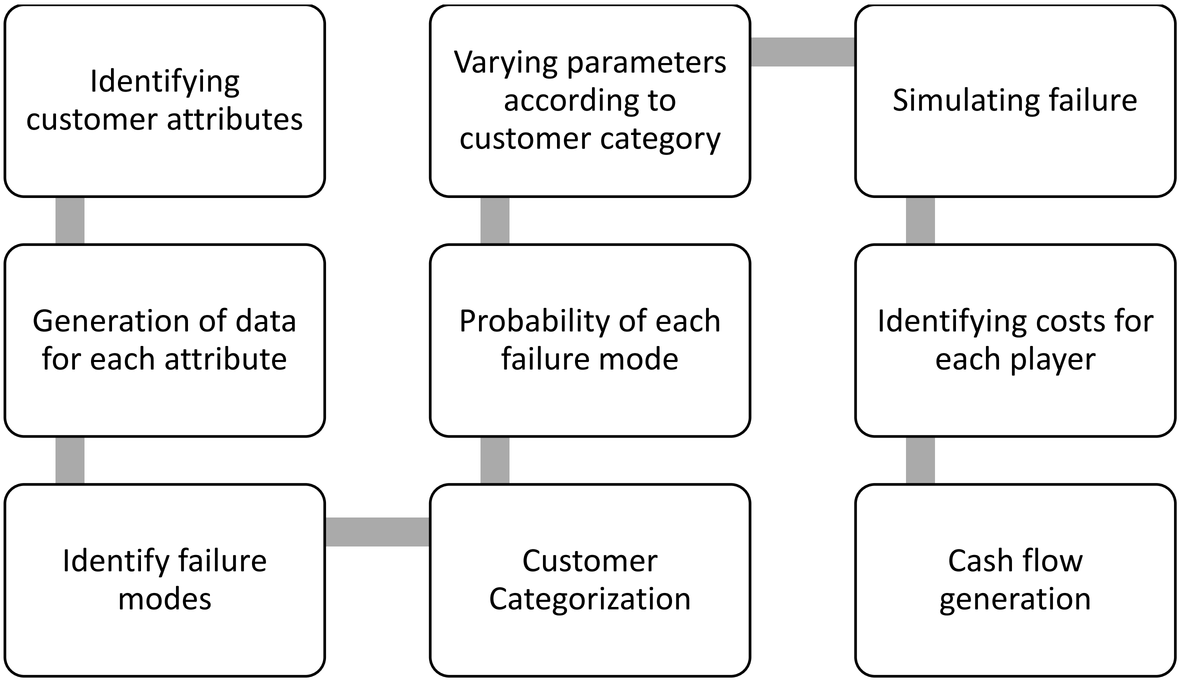


Figure 3.1 Process flow chart for data simulation

Chapter 4

Customer Profiles: Data Simulation and Insights

4.0 Introduction

As outlined in chapter 3, the first step is to generate customer related data and carryout necessary analysis. In this chapter, we'll focus on simulation and analysis of customer profiles the details of which are discussed in the following sections.

4.1 Customer Attributes

A customer's profile is a collection of different attributes. There are three different types of customer attributes present in the data. They are as follows-

Type I - Numerical: Value is any number greater than zero, for example Age = 22

Type II - Binary: Values are zero or one, for example for a male, Male = 1 & Female = 0

Type III - Scaled: Values are on a scale of 1-10, Driving skill on a scale of 1-10, 10 being the best

4.1.1 Customer attribute names and descriptions

The table 4.1 shown below lists the various categories of attributes which distinguishes the customer along with the name of the attributes and their respective descriptions. Each attribute has on the there are three types of values. For example, "Married" has type II value i.e. Binary suggesting that it can be either 0 or 1 only; "Age" has type I value i.e. Linear suggesting that it can have any value greater than zero; "Disc_scale_of_10" has type III value (Scaled) suggesting that it can have any value from 1-10 scale.

Table 4.1 Customer attributes names and their descriptions

Category	Name of attribute	Description
Serial Number	Serial_Number	Serial number in number format
Name	Customer_Name	Name in format: First Name <space> Last Name
Age	Age	Age (in years)
Gender	Male	Sex (1 if Male, 0 else)
	Female	Sex (1 if Female, 0 else) [Considered only two genders due to low population of other genders]
Relationship status	Living_together	Relationship status as written, 1 if True, 0 if False
	Married	Relationship status as written, 1 if True, 0 if False

Table 4.1 continued

	Widow/Widower	Relationship status as written, 1 if True, 0 if False [Considers male and female both]
	Divorced	Relationship status as written, 1 if True, 0 if False
Earning status	Supported	Earning status, 1 if not earning else 0
	Supporting	Earning status, 1 if contributes financially at home else 0
	Bread_Earner	Earning status, 1 if sole earner else 0
Type of Job	Entrepreneur	Type of Job, 1 if entrepreneur else 0
	Unskilled Worker	Type of Job, 1 if unskilled job else 0
	Skilled Worker	Type of Job, 1 if skilled job else 0
	Management	Type of Job, 1 if management job else 0
	Farmer	Type of Job, 1 if Farmer else 0
Education	Primary	Level of Education, 1 if Primary
	Middle	Level of Education, 1 if Middle
	Senior-Secondary	Level of Education, 1 if senior secondary
	UG	Level of Education, 1 if UG
	PG	Level of Education, 1 if PG
Location	City	Location, 1 if city else 0
	Mountain	Location, 1 if Mountain else 0
	Village	Location, 1 if Village else 0
Purpose	Rental Service	Purpose of vehicle, 1 if vehicle is given on rent else 0
	Work	Purpose of vehicle, 1 if vehicle is used for work commute else 0
	Hobby	Purpose of vehicle, 1 if hobby is the purpose else 0
Experience	number of years	Experience of two-vehicle vehicle driving in years
Weight	Kg	Weight in Kg
Height	meters	Height in meters
Maintenance Habits	Regular	Maintenance habits, 1 if regular else 0
	Occasional	Maintenance habits, 1 if occasional else 1
Maintenance Enthusiasm	Passionate	Enthusiasm towards maintenance, 1 if Passionate else 1
	Normal	Enthusiasm towards maintenance, 1 if Normal else 1
Distance	Km/day	Average distance on vehicle in Km/day
Duration	Hours	Average usage duration per day of vehicle in Hours
Discipline	Disc_Scale_of_10 (10-best)	Riding discipline, Scale of 1-10
Pillion	Yes	Usually drives with a pillion, 1 if yes else 0
	No	Usually drives with a pillion, 1 if no else 0
Refueling habits	Always full tank	Refueling habits, 1 if effort is towards filling the tank full else 0

Table 4.1 continued

	more than half	Refueling habits, 1 if effort is towards filling the tank more than half else 0
	Refill only when empty	Refueling habits, 1 if effort is towards filling the tank only when almost empty else 0
Reporting	Scale of 10	Complaint reporting habit on a scale of 1-10 (10- reports immediately)
Spending outlook	Conservative	Outlook towards spending money on bike- Conservative 1 else 0
	Good	Outlook towards spending money on bike- Good 1 else 0
Income	0-5 Lakhs	Income bracket, 1 if as written else 0 (Amount in LPA)
	5-10 Lakhs	Income bracket, 1 if as written else 0 (Amount in LPA)
	>10	Income bracket, 1 if as written else 0 (Amount in LPA)
Religion	Hindu	Religion, 1 if Hindu else 0
	Islam	Religion, 1 if Islam else 0
	Sikh	Religion, 1 if Sikh else 0
	Other	Religion, 1 if Other else 0
House	Own House	1 if Own house, else 0
	Rented House	1 if Rented house, else 0
Cars	No Cars	Number of cars, 1 if No cars else 0
	1 Car	Number of cars, 1 if 1 car else 0
	>=2 Cars	Number of cars, 1 if >=2 cars else 0
Zone	North	Zone in India, 1 if North else 0
	South	Zone in India, 1 if South else 0
	East	Zone in India, 1 if East else 0
	North-East	Zone in India, 1 if North east else 0
	West	Zone in India, 1 if West else 0
Other Bikes	Yes	Bike from other brands, 1 if yes else 0
	No	Bike from other brands, 1 if no else 0
Family Members	Male adults	Family members, 1 if Male adults, 0 if no Male adults
	Female adults	Family members, 1 if female adults, 0 if no female adults
	Children	Family members, 1 if children present, 0 if no children present

4.1.2 Customer attributes' value generation

The table 4.2-2 below gives information regarding the generation of values for the customer attribute. For each category a pdf was selected, and parameters were set according to a guesstimate based on limited information available. As can be seen from the table, most of the values are mentioned in percentage. This percent represents the probability of an attribute compared to the total attributes in the that attribute's category. For example: If category is gender then the probability of male customer is 0.98 whereas female is 0.02. For linear variables, normal random distribution and the parameters were taken according to general trend data available over the internet.

Table 4.2 Customer attributes names and their descriptions

Category	Distribution	Name of attribute
Serial Number	-	Serial_Number
Name	-	Customer_Name
Age	Normal distribution (μ =35, σ = 15, min = 18, max = 70)	Age
Gender	98%	Male
	2%	Female
Relationship status	30%	Living_together
	63%	Married
	5%	Widow/Widower
	2%	Divorced
Earning status	25%	Supported
	50%	Supporting
	25%	Bread_Earner
Type of Job	20%	Entrepreneur
	15%	Unskilled Worker
	25%	Skilled Worker
	30%	Management
	10%	Farmer
Education	20%	Primary
	40%	Middle
	25%	Senior-Secondary
	10%	UG
	5%	PG
Location	50%	City
	20%	Mountain
	30%	Village
Purpose	30%	Rental Service
	50%	Work

Table 4.2 continued

	20%	Hobby
Experience	Normal distribution [μ =4 years, σ = 1 year,(Age - Experience) >= 18, Experience >=0]	number of years
Weight	Normal distribution [μ =80 kg, σ = 10 kg, min = 40, Max = 140 kg]	Kg
Height	Normal distribution [μ =1.75 m, σ = 0.5 m, min = 1.5, Max = 2 m]	metres
Maintenance Habits	40%	Regular
	60%	Occasional
Maintenance Enthusiasm	60%	Passionate
	40%	Normal
Distance	Normal distribution [μ =80 km, σ = 4 km, min = 0.5]	Km/day
Duration	Normal distribution [μ =1 hr, σ = 0.5 hr, min = 0.2]	Hours
Discipline	Normal distribution [μ =7, σ = 2, min = 0,max = 10]	Scale of 10 (10-best)
Pillion	30%	Yes
	70%	No
Refueling habits	20%	Always full tank
	40%	more than half
	40%	Refill only when empty
Reporting	Normal distribution [μ =7, σ = 2, min = 0,max = 10]	Scale of 10
Spending outlook	70%	Conservative
	30%	Good
Income	60%	0-5 Lakhs

Table 4.2 continued

	35%	5-10 Lakhs
	5%	>10
Religion	30%	Hindu
	10%	Islam
	40%	Sikh
	20%	Other
House	60%	Own House
	40%	Rented House
Cars	60%	No Cars
	20%	1 Car
	20%	>=2 Cars
Zone	25%	North
	10%	South
	10%	East
	30%	North-East
	25%	West
Other Bikes	40%	Yes
	60%	No
Family Members	30%	Male adults
	30%	Female adults
	40%	Children

4.2 Data Insights from Customer Profiles

Customer profiles proves to be a highly valuable data for the data analysts. It is imperative to take advantage of the customer data and derive valuable insights from it. One of the many famous techniques is customer segmentation. In this section we'll explain what customer segmentation is and how to derive it from the given data.

4.2.1 Segmenting Customers

Customer Segmentation is a method to segment customers such that customer with similar profiles in a specific way belong to same category. K-Means Clustering algorithm helps in achieving this segmentation. The algorithm was applied to 400 customers with number of clusters = 4. To form the clusters, customers demographics (age, race, religion, gender, family size, ethnicity, income, education level), geography (where they live and work), psychographic (social class, lifestyle and personality characteristics). Since the data generated in the previous section, had all these factors, it became very easy to form the clusters.

4.2.2 Process flow for Customer Segmentation

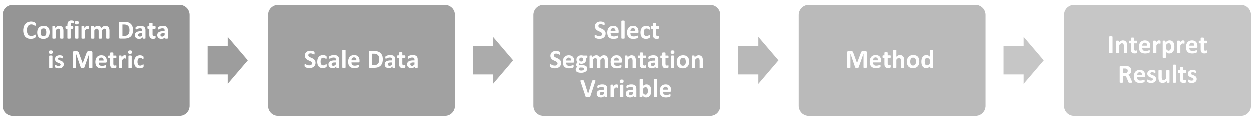


Figure 4.1 Process for Customer segmentation.

The data generated in section 4.2.2 is in metric system so the first step is taken care of. Further steps are taken care in the code snippet shown below.

4.2.3 Results

df1 – Dataset containing customer profiles data

```
scaled.df1 <- scale(df1) ## Scaling
df1 <- data.frame(matrix(c(df[,"Age"],df[,"Unemployed"],

df[,"Entrepreneur"],df[,"Unskilled.Worker"],df[,"Skilled.Worker"],
                           df[,"Management"],df[,"Farmer"], df[,"City"],
                           df[,"Mountain"],df[,"Village"],
                           df[,"Kg"],df[,"Km.day"]),nrow = nrow(df), ncol =
12))
## Selecting segmentation variables
k1 <- kmeans(df1,centers = 4) # K-Means Clustering algorithm used
```

As can be seen from the code snippet, K-Means clustering algorithm is used for segmentation. For detailed description of K-Means clustering please refer to 2.3.1.

As seen from figure 4.2, the customers formed are differentiated with the help of colors (there is no relation for colors of clusters in A with colors of clusters in B). The rectangular boxes represent the means of clusters. The axes are chosen randomly just for presentation purposes, we can plot the graph by picking any pair of combination of the selected attributes.

4.3 Conclusion

In this chapter, we provided the method to segment customers, and obtained customer segments using the K-Means algorithm. This insight is very valuable if we wish to make any changes to the current services, we'll make changes according to the segments generated. The utility of this exercise will be shown in Chapter 8, where we'll use the information obtained in this section and use it to optimize a service

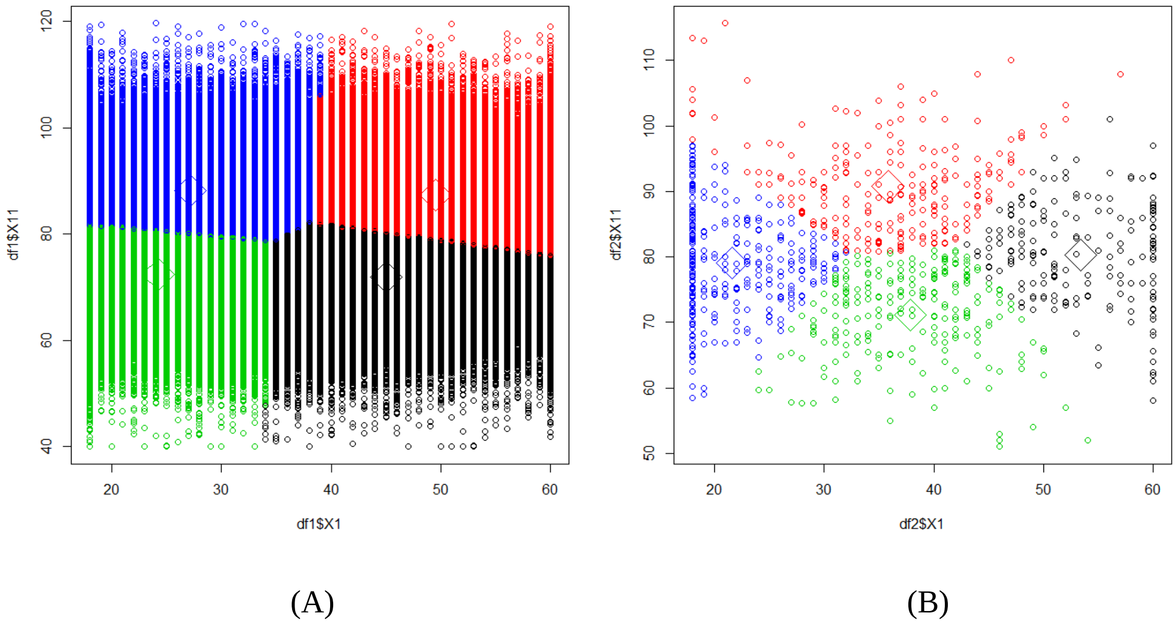


Figure 4.2 K-Means clustering on 100k customers (A) and 400 customers (B) [X axis – Age, Y axis - Weight]

.

Chapter 5

Failures: Data Simulation and Insights

5.0 Introduction

To simulate the failures, a basic understanding of the common types of failure occurring in the vehicle is imperative. To study the same, Royal Enfield Bullet 350cc model was selected. We'll first find out the major types of failure occurring in the vehicle. Then we'll categorize the customers in a way that the average failure rate for each category is different, this will help us in represent the real-life situation better. Then, for each category we'll simulate failures and derive insights from this simulated data.

5.1 Failure Modes

Various local mechanics, who are involved in the repairing of such vehicles, were interviewed to gain better understanding of the types of failure occurring in the vehicle. Based on these interviews, a table has been formulated as shown in table 4.3-1. The number of mechanics interviewed for this investigation were few, hence the data may not be very accurate or may not cover all the types of failure. Therefore, we consider this table only for representation purpose, and not a true table. This representation table will help us in formulating our system of analysis, and when applying the system in a real-life scenario effort should be made to formulate the failure modes table such that all the failure modes are included.

Table 5.1 Failure Modes

Type	Failure
1	Chassis break
2	Breaks
3	Electrical
4	Spokes breakage
5	Engine noise
6	Carburetor
7	Chain set break

5.2 Customer categorization

The main aim to simulate the data was imitate the actual market data, because through the simulation we aim to make predictions and analysis using machine learning algorithms. Therefore, the simulated data of failure should be similar to what we will observe in real-life scenario. The failure rate for a customer depends on many factors, some customers are more prone to facing failures in bike because they live in a region where the roads are patchy or proper maintenance of vehicle is not done whereas someone who lives in a city with good conditions and carries out proper maintenance of bike will have lower probability of vehicle failure. To imitate the same pattern in our failure data, customer categories were made based on some of the attributes of their profile, these categories are named as A, B, C and D. Following procedure was followed to categorize the customers:

Step 1: Select top 5 attributes which may play a major role in affecting the failure in the customer's vehicle

Result:

1. Experience

2. Usage

3. Terrain

4. Maintenance Habit

5. Riding Discipline

Step 2: Construct a failure score matrix with number of rows = number of customers and number of columns = 5 (1 column for each attribute)

Result:

Failure score matrix ideated

Step 3: Divide each attribute into different sections

Result:

1. Experience divided into 0-0.5 years, 0.5-1.5 years, 1.5-2.5yrs, 2.5-3.5 years, 3.5 years and above

2. Usage divided into 0-3 km/day, 3-5km/day, 5-8km/day, 8-11km/day and 11km/day and above

3. Terrain divided into City, Village, Mountain

4. Maintenance habits into Poor, Average and Good

5. Riding discipline into 5 equal sections in 1-10 scale

Step 4: Allot Failure Score Number to each section of each attribute

[Failure Score Index (FSI) is an integer number ranging from 1-5, this number is proportional to the probability of failure due to the attribute in a section. For example, FSI for Experience 0-0.5 should be 5 and for experience greater than 5 should be 1].

Result:

The final Failure Score Index matrix generated as shown in table 5.3-1

Table 5.2 FSI with Attributes

		Failure Score Index (FSI)				
		1	**2**	**3**	**4**	**5**
Attributes	Experience	3.5 years & above	2.5-3. years	1.5-2.5 years	0.5-1.5 years	0-0.5 years
	Usage	0-3 km/day	3-5km/day	5-8km/day	8-11km/day	11km/day
	Terrain	City		Mountain		Village
	Maintenance Habits	Good		Average		Poor
	Riding Discipline	8-10 out of 10	6-8 out of 10	4-6 out of 10	2-4 out of 10	1-2 out of 10

Step 5: Sum FSI over each attribute for each customer and find Final Failure Score Index

Result:

Final Failure Score Index (FFSI) = number between 5-25

Step 6: Allot customer categories A, B, C & D according to table shown below. From the table we can see that, if a customer has FFSI = 7, then she/he will belong to category A since the 5 < FFSI < 10. Similarly, for each customer, the categories can be found out.

Table 5.3 Customer Category according to FFSI

FFSI		**Category**
Lower limit	**Higher limit**	
5	10	A
10	15	B
15	20	C
20	25	D

5.3 Simulation of Failures

By now we have generated customer attributes and categorized them according to a defined rule (section 5.2). The next step is to simulate failures for each customer. The failures are generated month-wise for a period of 24 months (since the warranty period is 24 months or 20k km, assuming the manufacturing company will have failure data of at least 24 months for all vehicles). To simulate the failures, Weibull probability distribution function (pdf) was taken. The Weibull parameters were different for each customer category. For example, customer category A has lower FFSI hence the parameters were set such that the MTBF was more whereas for D the MTBF was less since it has

Here is a tabulated form of parameters for each customer category

Table 5.4 Weibull Parameters for each customer category

Category	β	η
A	2	100
B	2	80
C	2	50
D	2	30

Next step was to simulate the failure based on the pdf. Each simulation generates a random number, this number signifies number of days to failure. This failure is then registered in the failure matrix. All constraints are taken care in the code written in the appendix. A small snippet of the code is attached to provide clear understanding regarding the simulation.

```
if(A(i))
        while(Time <= NMonths  )
            mttf = wblrnd(eta_A,beta_A);
            Time = Time + mttf/30;
            Time = round(Time);
            if(Time <24 && Time > 0 )
               Failure(i,Time) = 1;
            end
        end
end
```

Description of variables:

A = Binary, 1 if customer belongs to category A, else 0

NMonths = Number of warranty months

wblrnd = Matlab function to generate Weibull random numbers with given parameters

Time = Time for failure in months

mttf = Random number generated through pdf (in days)

Failure = Failure matrix (number of rows = number of customers, number of columns = NMonths)

5.4 Data Insights from Failure Data

By now we have, customer profiles and the failures occurring in the first twenty-four months. We can use this data to derive various insights. In this section we'll use linear regression which will give us relation between various variables in the data. Further, we'll also aim to predict the time of failure for any new customer.

5.4.1 Linear Regression on Failure Data

Various linear regression can be performed on the failure data, for example, as shown in figure 5.1. On the x-axis, we have age of the customers whereas on the y-axis, we have total failures in 24 months.

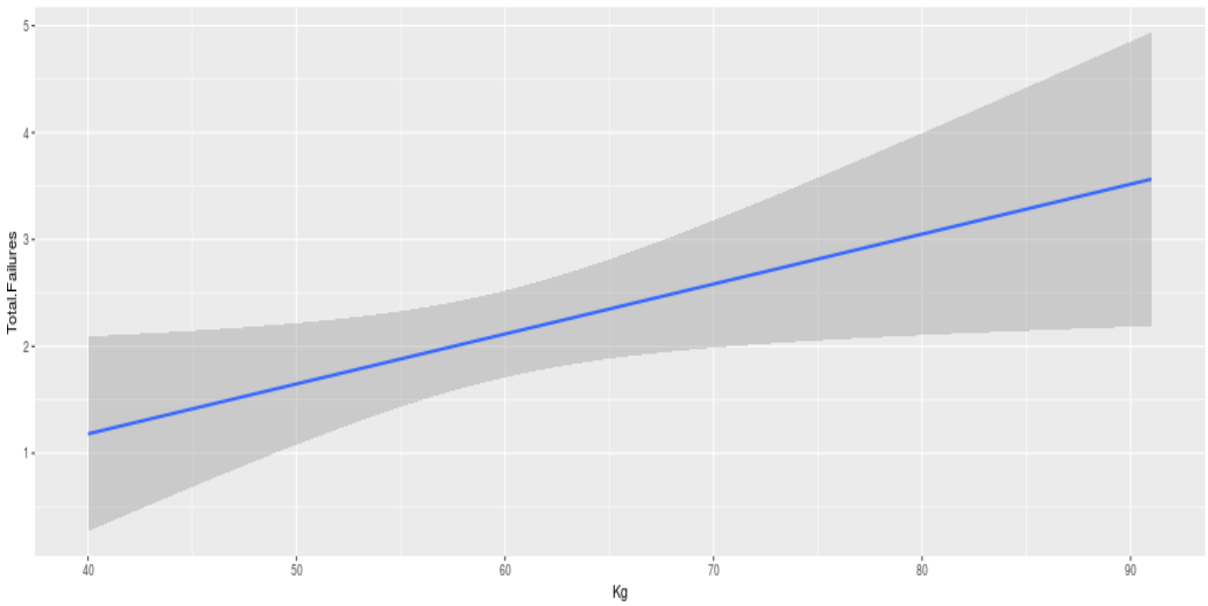


Figure 5.1 Total failures with age in initial 24 months

As can be seen from the figure 5.1, The grey area shows the 95% confidence interval while the blue line is the linear regression line. The total number of failures are increasing for customers with higher weights. This shows us the impact of weight on the reliability of vehicle, although we should keep in mind that weight can be one of the many factors that affects the failures. Similarly, we can change the variable on the x-axis and derive valuable insights.

5.4.2 Prediction of failures

We can use various machine learning algorithms to predict failure month for any new customer by training the algorithm on the training data (data generated in chapter 4 and section 5.4). The following algorithms were used in predicting the failures.

To run the prediction algorithm, we took failure data for 4000 customers. The attributes of customer profiles were the predictor variables whereas failure in a specific month was the dependent variable to be predicted. 80% of the customer profiles were random selected from the 4000 customers to make the training data set. The rest 20% acted as the test data set. The results obtained from the prediction algorithms were then tested against the test data set and the results were tabulated as shown in table 5.5-1. For representation purposes, the dependent variable is taken to be failure in 20th month after the date of purchase. This can be changed to any of the 24 months.

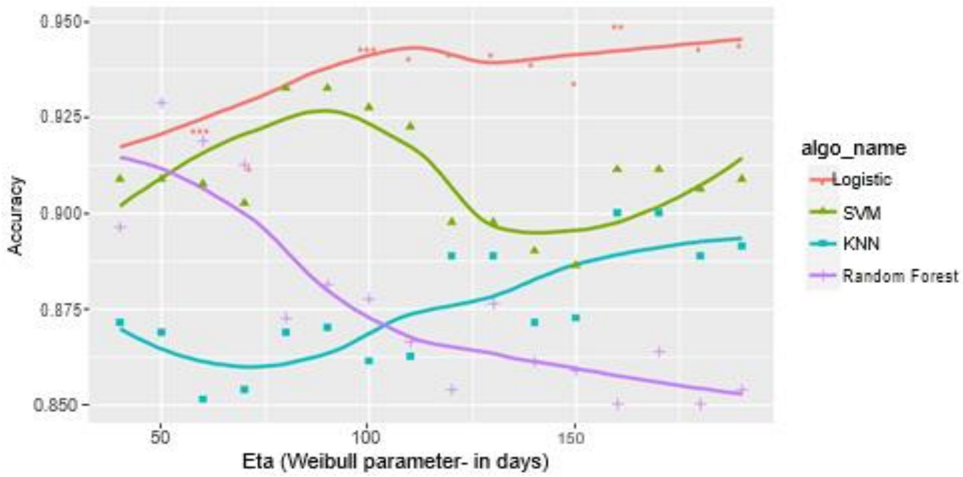
The following table summarizes the results obtained for predictions of failure in 20th month after the date of purchase.

Table 5.5 Performance of various algorithms

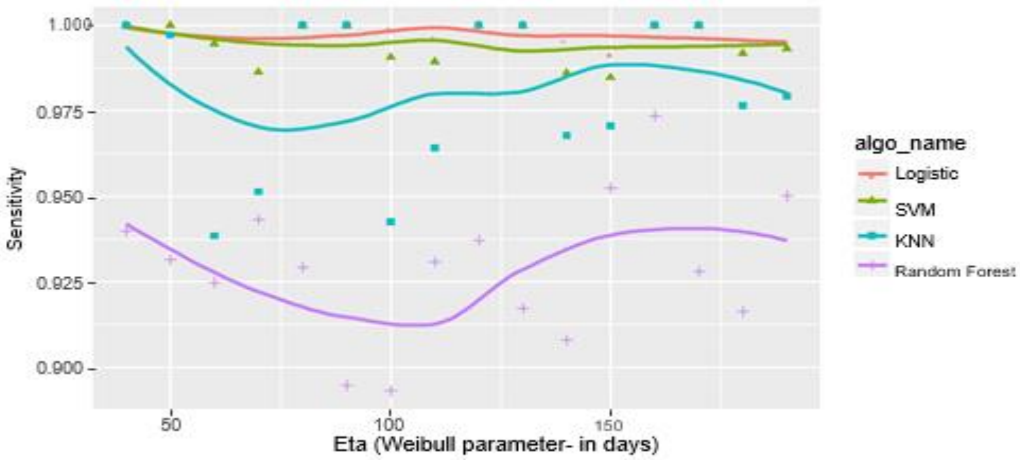
Algorithm Name	Accuracy	Sensitivity	Specificity
Logistic Regression	0.8464	0.9547	0.2069
SVM	0.8552	1	0
KNN	0.8552	0.9679	0.1897
Random Forest	0.843	0.9811	0.0201

As seen from the above table, these algorithms' performance is very different when comparing on different indices. To predict a failure, meaning that to predict 1 when actually the value is 1. This rate is captured by specificity, which is in the last column of the above table. As we can see, Logistic regression and KNN both perform considerable well on this index. Further the performance of these algorithms was tested on different number of customers and different failure rates. The results are shown figures shown below. To vary the failure rate, the Weibull parameter η (in days) was changed and to vary the data size, number of customers were increased.

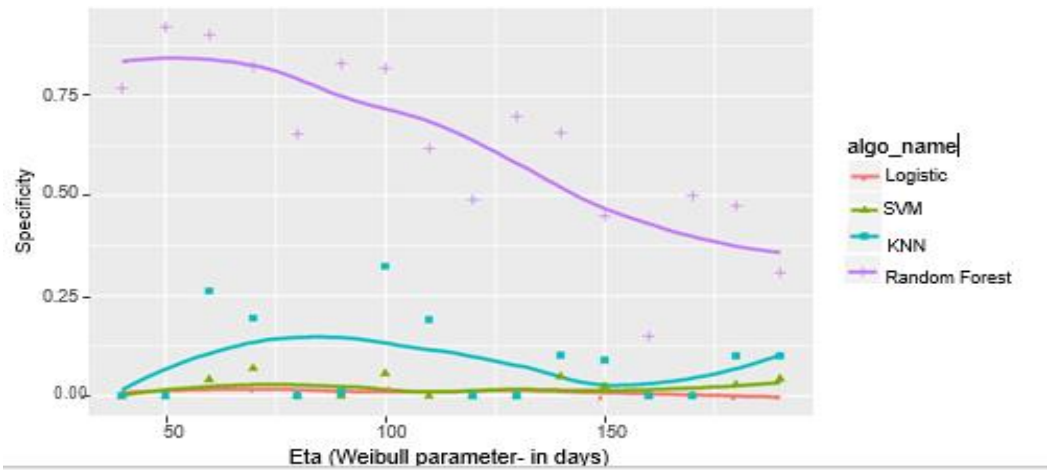
Figure 5.2 shows the variation of various performance indices with changing η (in days). Along the x-axis the value of η is increasing, implying that the MTTF is decreasing (not necessarily proportional) and the failure rate is decreasing. This means that if we move from left to right on the x-axis the number of failures occurring are less. For logistic regression the accuracy increases, but sensitivity and specificity remain almost constant. We can see that the performance of Random forest algorithm is best on the specificity index when the failure rate is more. We should keep in mind that the most important index for our analysis is specificity. Although the sensitivity of the algorithms other than random forest seems to be best but we can't judge based on sensitivity. Since, if an algorithm predicts all the values to be zero regardless of the data, then it'll have a sensitivity of 1, and since the training data has large number of zeros hence if an algorithm predicts all values to be zero, it may seem good on accuracy and sensitivity, but the algorithm is not efficient. Hence, we shall not be considering accuracy and sensitivity as a performance comparison parameter while we vary η. While looking at figure 5.3 we can see that the maximum values of specificity is observed for logistic and KNN algorithm, as the customers are increasing the performance of KNN and logistic is also increasing on specificity scale. That is for a fixed value of η we observe KNN and logistic are improving with number of customers. Hence if we have large number of customers we might rely on the results obtained from KNN and logistic.



(A)



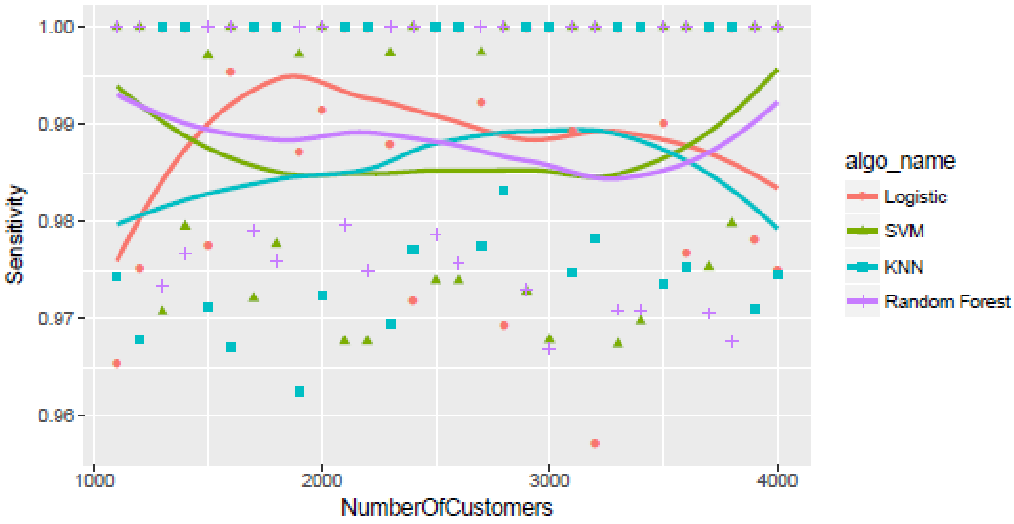
(B)



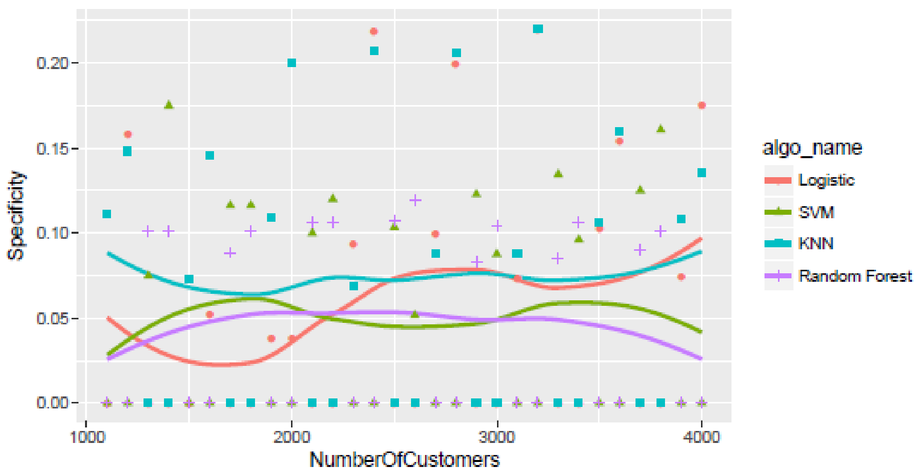
(C)

Figure 5.2 Variation of Accuracy (A), Sensitivity(B) and Specificity(C) with Eta (Weibull-parameter)

(A)



(B)



(C)

Figure 5.3 Variation of Accuracy (A), Sensitivity(B) and Specificity(C) with Eta (Weibull-parameter)

5.5 Generation of Cash Flow

There are three players in this system, first one is manufacturer, second one is service provider and third one being the customers. For each type of failure, all these players have some cost. For analysis purpose, these prices are decided randomly, but the costs are increasing according to their severity. For example, a chain breakage may have lower cost to the customer and an engine failure may have higher cost. These costs have been summarized in table 7.6-1. Positive values represent the costs payed and negative values represent the money earned.

Table 5.6 Costs for each type of failure

Costs (INR)		**Type of Failures**						
		1	**2**	**3**	**4**	**5**	**6**	**7**
Player	Manufacturer	7000	300	3500	700	900	1200	200
	Service Provider	-4000	-3400	-2000	-1000	-500	-400	-200
	Customer	700	300	500	400	300	200	100

The table considers costs of replacement items for the manufacturers, this is the reason behind the non-zero sum of all costs. For example, in type 1 failure, cost to manufacturer = INR 7000; cost to service provider = INR -4000; cost to customer = INR 700. The sum of these values comes out to be INR 3700, this cost must be the price of the replacement item.

These values of costs were used to calculate cash inflow and outflow each month based on the type of failure occurring (if occurring) to each customer. This resulted in a cost matrix which help in formulating the cash flow.

5.6 Conclusion

In this chapter, the method to generate customer failures while relating the failure rates to the customer profiles was shown. The primary reason to simulate data using the method shown was to emulate the real-life situation where failure rate of a customer would largely depend on his/her demographic, geographic and psychographic characteristics. We leverage the available of customer data of all these characteristics and simulated failures. Once the failures were simulated, we derived insights from the data and weighed upon the importance of such insights while taking business decisions. We also use various algorithms to make predictions on new customers and validated results using the cross-validation method. Various indices to compare the performance of the algorithm were introduced and we saw that Random forest works best when the failure rate is more, but as the number of customer increases KNN and logistic algorithm fare better.

Chapter 6

Ratings: Data Simulation and Insights

6.0 Introduction

Customer can rate the service provided to them on different factors and also on an overall basis. We can utilize the data obtained from the feedback obtained from the customer in the form of ratings and try to improve the customer satisfaction. To improve the customer satisfaction, either we can increase the level of service of all the factors or we can try to optimize customer satisfaction with costs by shifting customers to the type of service they prefer according to their profile. This chapter describes various terminologies used for such optimization and the method to carry out the optimization.

6.1 Customer Perceived Value (CPV)

Customer Perceived Value evaluates the level of satisfaction through the services or product. In this chapter, we aim to capture CPV based on the services provided during warranty. In real life scenario, this value is provided by a customer at the end of the service. The values lie between 1-10.

To simulate these values the formula written below is used.

$$CPV = \sum_{i=1}^{N} w_i R_i$$

where,

w_i = weight of factor i

R_i = Rating given by customer on factor i

This value is calculated for each customer.

The main aim of the project is to increase the customer perceived value while keeping the costs low, it is very imperative to create a model for the same. In this regard, ratings play a major role. They help the service provider to evaluate their levels of service and improve accordingly. Our focus is towards warranty service and hence all the work has been focused on improving the experience of customer if he/she needs to avail the warranty. This chapter briefly explains the steps taken while generating the data.

6.2 Factors affecting Customer Perceived Value (CPV)

The CPV has been divided into various factor, each factor has its own effect on the CPV. These factors have been identified based on the flow of warranty process. The table 6.3-1 below shows the list of various factors identified.

Table 6.1 Factors affecting CPV

Serial Number	Factor Name	Description
1	Failure Rate	How many times have the same failure occurred per month
2	Severity	The severity of failure
3	Effort_initial	Effort required to lodge a complaint
4	First_Action_Time	Time taken to accept complaint and take first action
5	Online_Call_Support	Online and call support
6	Staff_Behavior	Staff behavior
7	Effort_during	Effort required once the warranty process has started
8	Professionalism	Professionalism of mechanics
9	Spare_Availibility	Availability of spares
10	Total_Time	Total time for complaint redressal
11	Quality	Quality of parts replaced or repaired
12	Warranty_Rep	Warranty of repaired replaced parts
13	Cost	Cost to customer
14	Outlook	Outlook of customer after service
15	Transperency	Transparency in billing

6.3 Rating Generation

For each factor the customer can give a rating from 1 to 10. These ratings signify the level of satisfaction on a factor. The ratings are proportional to the level of satisfaction. A rating of 1 for a factor means that the customer is least satisfied on that factor, while a rating of 10 signifies that the customer is fully satisfied on that factor.

For the ratings generation, random uniform probability distribution has been used. For a customer, the values of ratings is an integer from 1 to 10. For the purpose of analysis, these values have been simulated whereas in the real world, these ratings shall be taken from the customer in the form of a feedback after the warranty service. A snippet of the rating matrix generated is attached in table 5.3-1 below. Example customer with serial number 5, has given rating of 7 on factor 8 i.e. professionalism of mechanics (from table 5.2-1). Meaning that he/she is moderately satisfied on this factor.

Table 6.2 Ratings based on factors given by each customer

Ratings	Factors															
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Customer Serial Number	**1**	6	2	9	2	3	8	4	6	2	2	6	1	2	6	4
	2	8	4	5	1	6	5	2	4	2	6	6	9	2	6	3
	3	10	7	5	10	7	10	4	7	4	7	9	2	4	1	4
	4	9	6	10	4	8	2	2	8	10	2	9	1	10	4	1
	5	3	2	3	10	1	9	3	8	2	7	6	1	3	1	2
	6	1	3	8	3	4	6	9	5	10	2	10	4	9	1	9
	7	3	5	10	7	7	6	2	2	10	2	3	4	9	10	9
	8	3	5	5	8	2	1	6	4	3	2	8	5	7	7	5
	9	10	1	6	7	10	5	8	2	8	8	5	3	5	2	3
	10	8	5	7	7	7	3	8	10	7	10	6	4	9	6	9

6.4 Weights Generation

Each customer has inherent biases towards services. These biases determine the overall satisfaction of the customer i.e. CPV. To simulate the same, a concept of weights for factor has been introduced, this will help us simulate the values of CPV in a more realistic manner. The picture will become clearer once the we formulate the CPV generation in section 5.5.

For the weights generation, random uniform probability distribution has been used. For a customer, the sum of these weights is equal to one.

A snippet of the weights matrix generated is attached in table 6.5-1 below.

Table 6.3 Weights of each factor based on customer bias

Weights	Factors															
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Customer Serial Number	1	0.1	0.1	0.1	0	0	0	0	0.1	0.1	0.14	0.07	0.07	0.05	0.09	0.04
	2	0.1	0	0.1	0.1	0	0.1	0	0.1	0	0.05	0.11	0.07	0.08	0.07	0.11
	3	0.1	0.1	0	0	0	0.1	0.1	0	0	0.12	0.09	0.08	0.06	0.13	0.02
	4	0.2	0.1	0.1	0.1	0.1	0.1	0	0.1	0	0.02	0.07	0.03	0.07	0.03	0.08
	5	0.1	0.2	0	0.1	0	0	0.1	0.1	0	0.1	0.13	0.14	0.01	0.02	0.08
	6	0.1	0.1	0.1	0	0.1	0.1	0	0	0	0.06	0	0.1	0.01	0.06	0.11
	7	0	0.1	0	0.1	0	0.1	0.1	0.1	0	0.08	0.09	0.11	0.05	0.09	0.04
	8	0	0.1	0	0.1	0.1	0	0.1	0	0.1	0.02	0.04	0.17	0.01	0.14	0
	9	0	0	0	0.1	0	0.1	0.1	0.1	0.1	0.05	0.11	0.13	0.02	0.03	0.02
	10	0.1	0.1	0.1	0.1	0	0	0.1	0.1	0.1	0.13	0.02	0.05	0.11	0	0.02

6.5 Level of Service (LOS)

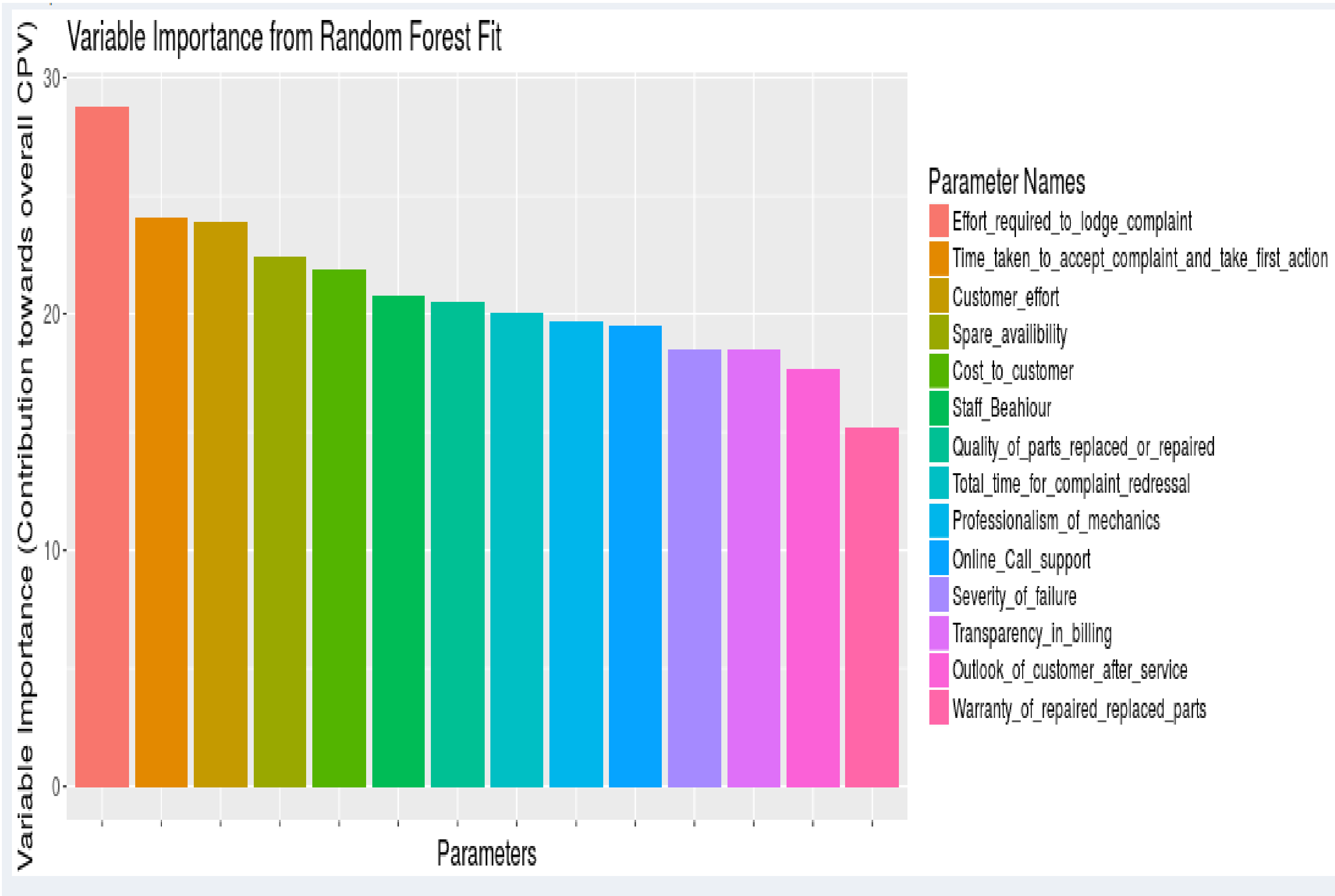
Each customer faces different levels of service since the service provider various from customer to customer. Each factor of the service can have different levels. These levels are captured from a 1 to 5 scale, 1 belonging to the worst level of service and 5 belonging to the best level of service. To gauge the level of service, internal audit can be performed on different service provider and after the audit the service providers can be rated from a scale of 1 to 5. This will help us further when we proceed with our algorithm to optimize customer satisfaction with the costs. For now, the values of LOS are randomly simulated for each customer and each factor. The table for LOS will be similar to the table 6.5-1 but with the values in the table ranging from 1 to 5 instead of 0 to 1.

6.6 Data Insights from Ratings Data

To analyze the ratings data, a specific type of plot was ideated. This will help us in analyzing the average rating given by different category of customers on different factors of service. The category of customers can be chosen as per the choice of the analyst. Efforts were made to keep these categories as much independent as possible. For example, a plot generated for analyzing customers with categories based on location i.e. Category 1 = Living in City area, Category 2 = Living in Village area, Category 3 = Living in Mountain terrain area. The average ratings of each category for specified factors is plotted as shown in Figure 6.1. It can be inferred from this specific plot that customers living in the mountain area are least satisfied when comparing on the factor of effort to lodge a complaint and severity of failure. Meaning that customer from mountain area are facing difficulty in lodging a complaint and the failures are more severe, but if on an overall scale they are more satisfied than the other two category customers. The pattern seen here may not be very logical since the data generated for ratings is very random and regardless of customer profiles, but in real life such plots can be very helpful and can prove to be very helpful in driving business decisions. All the categories of the x-axis can be changed, and the rating types can also be changed according to the choice of analyst.

We can also derive variable importance values for each factor on CPV using random forest algorithm. This will help us in knowing average weightage given to factors by customers. We will use this insight in Chapter 9. The variable importance plot for all customers is shown in figure 6.2. It can be seen that for the given data set average weight given to factor "effort_required_to_lodge_complaint" is the highest, while it is lowest for factor "warranty_of_replaced_parts".

Figure 6.1 Variable importance plot



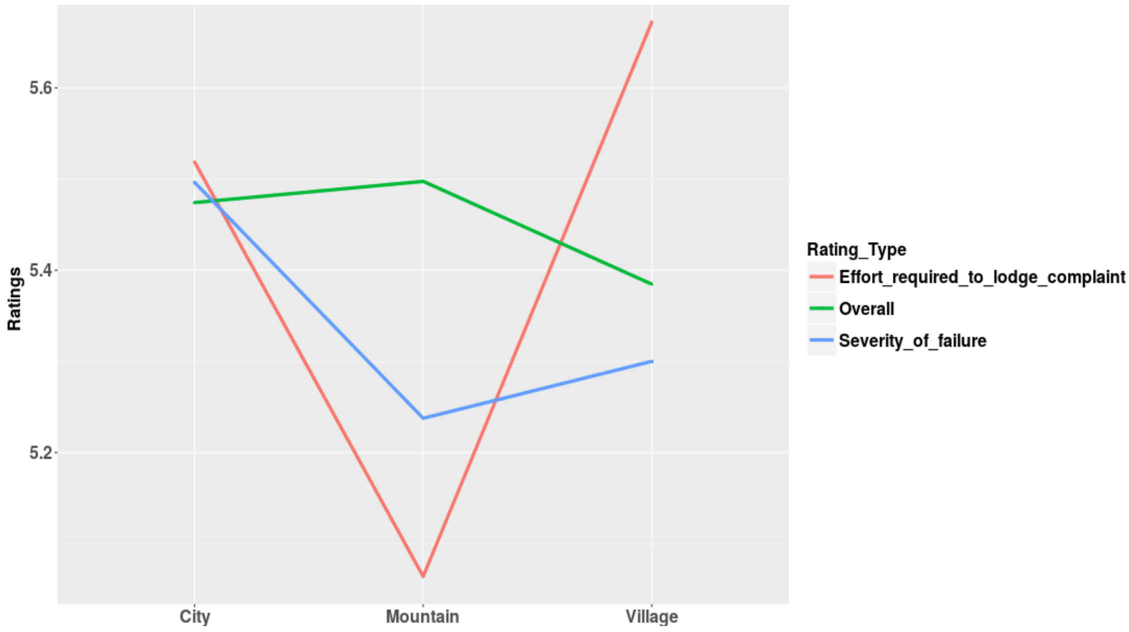


Figure 6.2 Rating analyzer plot

6.7 Conclusion

Rating values required for analysis in the further chapter are generated here. In real life scenario, only the rating values and CPV will be filled by the customer in the customer form. The weights table was generated only to calculate values of CPV giving us more realistic values and therefore imitating the real-life scenario. A specific plot for analyzing the ratings data was introduced, with an example shown in figure 6.1. These factors on the x-axis and rating type can be changed as per the need.

Chapter 7

Optimizing CPV with Costs

7.1 Introduction

We have simulated data related to LOS, CPV, Ratings etc. We can utilize this information to optimize the current distribution of services in such a way that the customer satisfaction is increased while keeping the costs low. We'll first develop a framework for the deriving an algorithm to do so, and then describe process flow of the algorithm. To deepen the understanding, we'll write a pseudo code which will guide us to apply the algorithm and derive results.

7.2 Framework

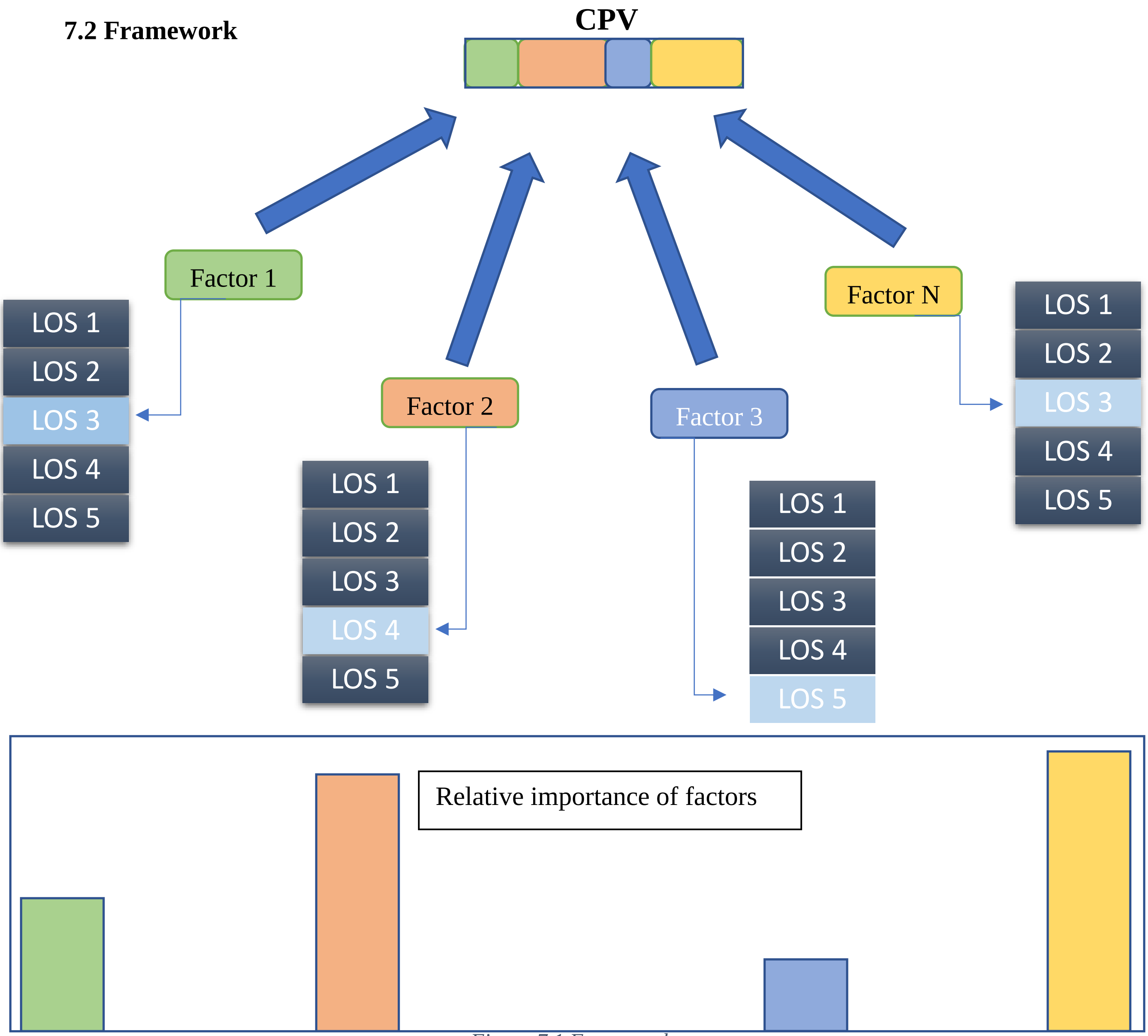


Figure 7.1 Framework

The above framework weaves the data we have collected/generated for the analysis in a systematic way. The value of CPV depends on various factors for example Factor 1, Factor 2, Factor3, …, Factor N etc. Each of the factor have different levels of service (LOS) from 1 to 5 as shown. For a particular customer Factor 1 has LOS = 3, Factor 2 has LOS = 4, Factor 4 has LOS = 1, …, Factor N has LOS = 3. Also, we can see from the corresponding color bars showing the relative importance of each of these factors. We also know that, as the LOS increases from 1 to 5 the average cost per customer increases. Now, the basic idea behind the algorithm is we should allot the LOS of a factor to a customer according to the relative importance. For example, in our illustration, Factor N has highest relative importance but the LOS being provided is only 3, therefore we can try to increase the level of service to 4 or 5 according to some defined rule. Similarly, Factor 3 has very low relative importance, but the LOS being provided is 5, therefore we can try to decrease the LOS to lower levels.

7.3 Process flow

The figure shown below outlines the process flow of applying the algorithm. We'll use this to write a pseudo code for the algorithm in the next section.

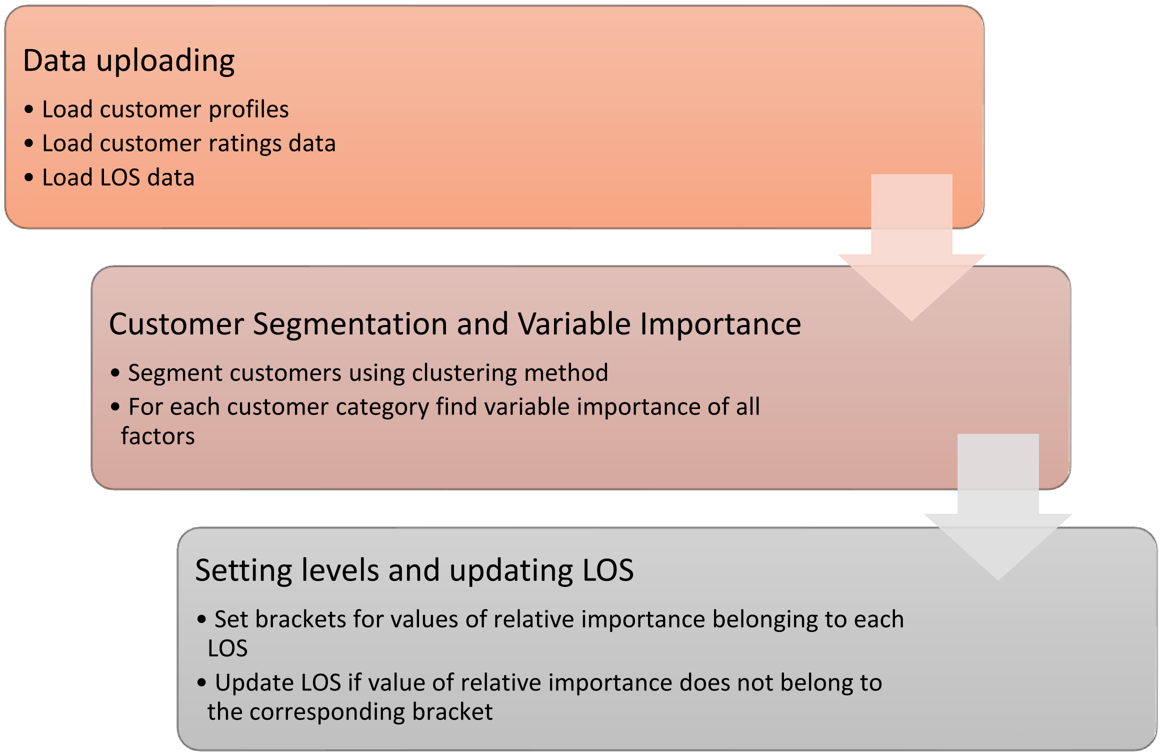


Figure 7.2 Process flow of implementing algorithm

7.4 Pseudo Code

Below is the pseudo code for the algorithm, comments are written with “#” symbol in front. Detailed code for the same can be found in Appendix III.

```
Begin
##### Data loading #######
Read Customer profile data
Read Customer ratings data
Read LOS data
##### Clustering ########
Apply K-Means clustering to generate 4 different customer segments – A, B,
C, D
##### Variable importance #####
Start For loop (each customer category)
Find relative importance using random forest algorithm
End For loop
#### Updating LOS ####
Start 1st for loop (for all category)
        Start 2nd for loop (for all customers of the current category)
                Start 3rd for loop (for all factors)
                        If (LOS < 5 and importance of factor > 10)
                        Update LOS = 5
                        Else if (LOS <4 and 8<importance of factor <=10)
                        Update LOS = 4
                        Else if (LOS < 3 and 5<importance of factor <=8)
                        Update LOS = 3
                        Else if (LOS > 2 and 2.5<importance of factor <=5)
                        Update LOS = 2
                        Else if (LOS > 1 and importance of factor <=2.5)
                        Update LOS = 1
                End 3rd for loop
        End 2nd for loop
End 1st for loop
##### Finding initial and final costs #####
Find initial cost using old LOS
Find updated cost using updated LOS
```

7.5 Results

For each customer category we get a updated LOS, that is if a customer has been updated some other level of service in the updated LOS then we need to change it from the current LOS to update LOS. Let us look at an example from the snippet of current LOS and updated LOS.

Table 7.1 Initial LOS

Customer Number	Factors														
5	5	5	4	4	5	3	4	4	4	1	3	2	5	2	4
6	2	5	2	4	4	2	3	3	4	4	1	3	2	4	5
7	2	2	5	4	5	2	5	4	2	1	5	2	2	4	5
8	1	2	2	4	5	2	3	1	4	4	4	3	4	4	2
9	3	5	1	5	1	1	5	5	1	3	2	5	4	5	4
10	5	4	5	2	1	2	5	1	3	1	1	3	3	2	5

Table 7.2 Updated LOS

Customer Number	Factors														
5	2	2	4	4	5	3	4	4	4	1	3	2	2	2	2
6	2	5	2	4	4	2	3	3	4	4	1	3	2	4	5
7	2	2	5	4	5	2	5	4	2	1	5	5	1	4	5
8	1	2	2	4	5	2	3	1	5	2	4	3	4	4	2
9	3	5	1	5	1	1	5	5	1	2	4	5	4	5	4
10	5	4	5	2	1	2	5	1	3	1	1	3	3	2	5

As can be seen from the above table, for customer numbers 5 to 10, we have highlighted the changed values in the updated table. For example, in the first row two LOS were updated to level of 2 from 5, similarly a LOS = 3 is updated to LOS = 2. All these changes are leading to increase in customer satisfaction by providing services according to their choices.

We'll also look at the cost perspective using the table 7.5-3. The table shows cost per customer for each factor and each level of service. As we can see for a factor, the price increases as the level of service increases which is similar to what should happen in real-life. This trend is maintained in all factors although the values are randomly generated.

Such simulation was run on 400 customers and 15 factors with 5 levels. The initial cost was found out to be INR 3,39,866 whereas the updated cost was INR 3,60,947 which is only 6.2% higher than the initial cost but LOS has been arranged in a manner such that the customers are more satisfied. Actual rise/fall in the overall CPV will only be seen when the customers get to experience updated levels of service and submit a feedback after the service.

Table 7.3 Average cost for each factor and LOS

Avg Cost (INR)	LOS 1	LOS 2	LOS 3	LOS 4	LOS 5
Factor 1	42	48	56	65	73
Factor 2	42	47	57	65	71

Table 7.3 Continued

Factor 3	43	47	59	65	72
Factor 4	38	48	55	65	73
Factor 5	41	51	55	64	75
Factor 6	42	48	56	63	73
Factor 7	43	48	59	62	71
Factor 8	41	50	58	64	74
Factor 9	41	50	56	64	70
Factor 10	39	51	55	65	74
Factor 11	42	47	57	65	71
Factor 12	42	49	55	64	72
Factor 13	40	48	59	65	71
Factor 14	39	49	59	63	74
Factor 15	40	47	55	65	71

7.6 Conclusion

An investigation was done to define how the factors are affecting the CPV and then framework and process flow for implementing the algorithm were shown. The algorithm was applied to the generated data, this resulted in increase in the costs of service by 6.2% and the CPV is expected to increase since the customers will be receiving higher levels of service for their preferred factors in service.

Chapter 8

Summary & Future Work

8.1 Summary

The understanding of problem began through learning the basic concepts involved in the analysis i.e. learning the fundamentals of warranty, data analytics, machine learning. Then, the focused was shifted the literature review of warranty analysis which helped in gaining better understanding of the problem and solutions related to warranty analysis. More deeper understanding of reliability was gained and the same was applied to analyze the number of claims under warranty and costs of warranty. The main problem statement was defined which helped to narrow down the problem for more concentrated efforts. To understand the problem better, first a better understanding of market scenario was made through research on the market by finding out major brands, models and their respective warranty policies. Various forms of data were generated to imitate the real-life data from the industry. Different methods to analyze the data were shown. A simplified cost-effective system for the improving CPV was suggested and implemented on the generated data.

8.2 Gains of the Study

Lot of effort and time were invested to create a data which represents real life data, this understanding can be used while carrying out analysis in different fields as well. The system developed for analysis of the created data helped in gaining inferences from the data which can help in driving various industrial decisions. The system developed for cost effective optimization of CPV can help the company to increase the CPV. This can help in increasing the intangible assets of the company such as brand value.

8.3 Limitation of the Study

Following are the limitations to the study:

I. Data Quality – In the vehicles industry, although the amount of data is increasing but the quality of data being generated is not at par with the required quality. This will affect the application of our study since the data is the fundamental for the analysis and inference.

II. Data Volume – To increase the effectiveness of the system developed, the volume of data needs to be very high such that all the customers are included in the dataset. There are various regions, in the country where data generation is not possible due to multiple reasons. Also, the algorithms applied in the system work better as the data volume increases hence the volume of data affects the efficiency of the system developed.

III. Inefficient Auditing of Services – In section 6.4, the concept of LOS was introduced. The LOS values for the services are based on audit conducted by either the company or any third party. Since the company is assumed to be operational across India, it will be very difficult for audit of services to normalize the audit across such large geography. This inefficiency can prove hazardous and may result in absurd results.

IV. Customer Segmentation – The algorithm used for customer segmentation uses demographic, psychographic and geographic data from customer profiles. This segmentation concludes that behavior of the customers from a segment shall be similar, but there is a possibility of misclassification of some customers, which will result in inaccurate shifting of customers' services.

8.4 Scope for Future work

Future work would constitute implementing the changes suggested by the algorithm in chapter 7 to real-life situation and observe the reactions of customers. At the initial level, this can be implemented on a test-basis to small number of customers. If the outcomes are aligning with our expectations already established, then we can increase the scale of application. Also, the process of applying algorithms itself requires lots of different exercises, efforts can be made to improve the efficiency of each exercise to improve the results. There is possibility of application of more advanced machine learning algorithms such as Neural Networks and Xgboost which require large amount of data as compared to the algorithms used in the system developed in this report.

References

1. Blischke, W. R. and D.N.P. Murthy. (1991). 'Product warranty management—I. A taxonomy for warranty policies.' European J. Operational Research.
2. Murthy, D. N. Prabhakar, Blischke, Wallace R. (2006). 'Warranty Management and Product Manufacture'. The Springer Series in Reliability Engineering.
3. Jeffrey A. Robinson. Warranty Claims and Costs: Statistical Analysis of. Wiley Publications.
4. Kalbfleisch, J.D. and Lawless, J.F. (1991), "Regression models for right truncated data with applications to AIDS incubation times and reporting lags", Statistica Sinica, Vol. 1, pp. 19-32.
5. Chukova, S. and Dimitrov, B. (1996), "Warranty analysis for complex systems", in Blischke, W.R. and Murthy, D.N.P. (Eds), Product Warranty Handbook, Marcel Dekker, New York, NY, pp. 543-84.
6. Karim, M.R., Yamamoto, W. and Suzuki, K. (2001a), "Statistical analysis of marginal count failure data", Lifetime Data Analysis, Vol. 7, pp. 173-86.
7. Kalbfleisch, J.D., Lawless, J.F. and Robinson, J.A. (1991), "Methods for the analysis and prediction of warranty claims", Technometrics, Vol. 33, pp. 273-85.
8. Trindade, D.C. and Haugh, L.D. (1980), "Estimation of the reliability of computer components from field renewal data", Microelec. Reliab., Vol. 20, pp. 205-18.
9. Moskowitz, H. and Chun, Y.H. (1994), "A Poisson regression model for two-attribute warranty policies", Naval Research Logistics, Vol. 41, pp. 355-76.
10. A. Kleyner, P. Sandborn (2006), "Forecasting the cost of unreliability for products with two-dimensional warranties", Safety and reliability for managing risk, London
11. Majeske, K.D. (2003), "A mixture model for automobile warranty data", Reliability Engineering and System Safety, Vol. 81, pp. 71-7.
12. Crow A. Larry (1974), "Reliability Analysis for Complex, Repairable Systems"
13. Kim, H.G. and Rao, B.M. (2000), "Expected warranty cost of two-attribute free-replacement warranties based on a bivariate exponential distribution". Computers & Industrial Engineering, Vol. 38, pp. 425-34.
14. Vickie Lee Hill, Charles W. Beall and and Wallace R. Blischke, "A simulation model for warranty analysis". International Journal of Production Economics, 22 (1991) 131-140-131 Elsevier.

15. Zhiwei Chen, Tingdi Zhao, Shanshan Luo, And Yufeng Sun. “Warranty Cost Modeling and Warranty Length Optimization Under Two Types of Failure and Combination Free Replacement and Pro-Rata Warranty”.
16. Coursera.org.
17. Stefan Meinzer, Ulf Jensen, Alexander Thamm2, Joachim Hornegger, Björn M. Eskofier, “Can machine learning techniques predict customer dissatisfaction? A feasibility study for the automotive industry”. Artificial Intelligence Research. 2017, Vol. 6, No. 1.
18. T. Hill, “Machine Learning Techniques in Manufacturing”. Dell Research.
19. Wei Xie, “Optimal pricing and two-dimensional warranty policies for a new product”. International Journal of Production Research. ISSN: 0020-7543 (Print).
20. W.R.Bliscke “Mathematical Models for Analysis Of Warranty Policies”. Math/ Compur. Modeihng, Vol. 13, No. 7, pp. 1-16, 1990.
21. L. Lee, D. Dobler “Purchasing and Materials Management”, 1984.

Appendix 1

MATLAB Code for Customer Profile and Failure Generation

```
THE CODES IN THE FOLLOWING APPENDIX SECTIONS CAN BE FOUND OUT AT -

                    https://github.com/yashbaley/Thesis
```

```
Tic % Timer to calculate code running time
%% Reading excel file
clc; clear;
filename= 'SimulatedData_4000.csv';
sheet = 1;
NCustomers = 4000; %6 lakh customers
% [num, text, raw] = xlsread(filename,sheet) ;

%%
%%Generating ages
age = normrnd(35,15,[NCustomers,1]);
for i = 1:NCustomers
    if age(i)< 18
        age(i) = 18;
    elseif age(i) > 60
        age(i) = 60;
    end
end
% xlswrite(filename,age,sheet,'C3');
%%
%% Generating gender
Gender_column = rand(NCustomers,1);
Gender = zeros(NCustomers,2);

for i = 1: NCustomers
    if(Gender_column(i) < 0.98)
        Gender(i,1) = 1;

    elseif (Gender_column(i) >0.99)
        Gender(i,2) = 1;

    end
end
% xlswrite(filename,Gender,sheet,'D3');

%%
Marital_status_Column = rand(NCustomers,1);
```

```
Marital_status = zeros(NCustomers,5);
for i = 1:NCustomers
   if(Marital_status_Column(i)<0.3)
       Marital_status(i,1) = 1;
   elseif(Marital_status_Column(i)>0.3 && Marital_status_Column(i)<0.35)
       Marital_status(i,2) = 1;
   elseif(Marital_status_Column(i)>0.35 && Marital_status_Column(i)<0.9)
       Marital_status(i,3) = 1;
   elseif(Marital_status_Column(i)>0.9 && Marital_status_Column(i)<0.95)
       Marital_status(i,4) = 1;
   elseif(Marital_status_Column(i)>0.95 && Marital_status_Column(i)<1)
       Marital_status(i,5) = 1;
   end
end
% xlswrite(filename,Marital_status,sheet,'F3');
%%
Position_in_family_Column = rand(NCustomers,1);
Position_in_family = zeros(NCustomers,3);
for i = 1:NCustomers
   if(Position_in_family_Column(i)<0.25)
      Position_in_family(i,1) = 1;
   elseif(Position_in_family_Column(i)>0.25 &&
Position_in_family_Column(i)<0.75)
       Position_in_family(i,2) = 1;
   elseif(Position_in_family_Column(i)>0.75 &&
Position_in_family_Column(i)<1)
       Position_in_family(i,3) = 1;
   end
end
% xlswrite(filename,Position_in_family,sheet,'K3');

%% Occupation generation
Occupation_Column = rand(NCustomers,1);
Occupation = zeros(NCustomers,6);
for i = 1:NCustomers
   if(Occupation_Column(i)<0.2)
       Occupation(i,1) = 1;
   elseif(Occupation_Column(i)>0.2 && Occupation_Column(i)<0.4)
       Occupation(i,2) = 1;
   elseif(Occupation_Column(i)>0.4 && Occupation_Column(i)<0.45)
       Occupation(i,3) = 1;
   elseif(Occupation_Column(i)>0.45 && Occupation_Column(i)<0.60)
       Occupation(i,4) = 1;
   elseif(Occupation_Column(i)>0.60 && Occupation_Column(i)<0.80)
       Occupation(i,5) = 1;
   elseif(Occupation_Column(i)>0.80 && Occupation_Column(i)<1)
       Occupation(i,6) = 1;
```

```
    end
end
% xlswrite(filename,Occupation,sheet,'N3');
%%
%%Education
Education_Column = rand(NCustomers,1);
Education = zeros(NCustomers,5);
for i = 1:NCustomers
   if(Education_Column(i)<0.2)
       Education(i,1) = 1;
   elseif(Education_Column(i)>0.2 && Education_Column(i)<0.6)
       Education(i,2) = 1;
   elseif(Education_Column(i)>0.6 && Education_Column(i)<0.85)
       Education(i,3) = 1;
   elseif(Education_Column(i)>0.85 && Education_Column(i)<0.95)
       Education(i,4) = 1;
   elseif(Education_Column(i)>0.95 && Education_Column(i)<1)
       Education(i,5) = 1;
   end
end
% xlswrite(filename,Education,sheet,'T3');
%%
%%Affiliation
Affiliation_Column = rand(NCustomers,1);
Affiliation = zeros(NCustomers,3);
for i = 1:NCustomers
   if(Affiliation_Column(i)<0.8)
      Affiliation(i,1) = 1;
   elseif(Affiliation_Column(i)>0.8 && Affiliation_Column(i)<0.9)
       Affiliation(i,2) = 1;
   elseif(Affiliation_Column(i)>0.9 && Affiliation_Column(i)<1)
       Affiliation(i,3) = 1;
   end
end
% xlswrite(filename,Affiliation,sheet,'Y3');

%%
%%Location
Location_Column = rand(NCustomers,1);
Location = zeros(NCustomers,3);
for i = 1:NCustomers
   if(Location_Column(i)<0.5)
      Location(i,1) = 1;
   elseif(Location_Column(i)>0.5 && Location_Column(i)<0.7)
       Location(i,2) = 1;
   elseif(Location_Column(i)>0.7 && Location_Column(i)<1)
       Location(i,3) = 1;
```

```
        end
    end

    % xlswrite(filename,Location,sheet,'AB3');

    %%
    %%Purpose
    Purpose_Column = rand(NCustomers,1);
    Purpose = zeros(NCustomers,3);
    for i = 1:NCustomers
       if(Purpose_Column(i)<0.3)
          Purpose(i,1) = 1;
       elseif(Purpose_Column(i)>0.3 && Purpose_Column(i)<0.8)
           Purpose(i,2) = 1;
       elseif(Purpose_Column(i)>0.8 && Purpose_Column(i)<1)
           Purpose(i,3) = 1;
       end
    end

    % xlswrite(filename,Purpose,sheet,'AE3');

    %%
    %%Experience
    %[num, text, raw] = xlsread(filename,sheet) ;
    Experience_Column = normrnd(4,1,[NCustomers,1]);
    Experience = zeros(NCustomers,1);
    for i=1:NCustomers
        if(age(i,1)-Experience_Column(i,1)>18 && Experience_Column(i,1)>0)
            Experience(i,1) =  Experience_Column (i,1);

        end
    end
    % xlswrite(filename,Experience,sheet,'AH3');
    %%
    %%Weight
    Weight = normrnd(80,10,[NCustomers,1]);
    for i=1:NCustomers
        if(Weight(i)<40)
            Weight(i)= 40;
        elseif(Weight(i)>120)
            Weight(i) = 110;
        end
    end
    % xlswrite(filename,Weight,sheet,'AI3');
    %%
    %%
    %%Height
```

```
Height = normrnd(1.75,0.5,[NCustomers,1]);
for i=1:NCustomers
    if(Height(i)<1.5)
        Height(i)= 1.5;
    elseif(Height(i)>2)
        Height(i) = 2;
    end
end
% xlswrite(filename,Height,sheet,'AJ3');
%%
%%MHabits
MHabits_Column = rand(NCustomers,1);
MHabits = zeros(NCustomers,2);
for i = 1:NCustomers
   if(MHabits_Column(i)<0.4)
      MHabits(i,1) = 1;
   elseif(MHabits_Column(i)>0.4 )
       MHabits(i,2) = 1;
   end
end
% xlswrite(filename,MHabits,sheet,'AK3');
%%
%%Attraction
Attraction_Column = rand(NCustomers,1);
Attraction = zeros(NCustomers,2);
for i = 1:NCustomers
   if(Attraction_Column(i)<0.6)
      Attraction(i,1) = 1;
   elseif(Attraction_Column(i)>0.4 )
       Attraction(i,2) = 1;
   end
end
% xlswrite(filename,Attraction,sheet,'AM3');
%%
%%
%%Distance
Distance = normrnd(8,4,[NCustomers,1]);
for i=1:NCustomers
    if(Distance(i)<0)
        Distance(i)= 0;
    end
end
% xlswrite(filename,Distance,sheet,'AO3');
%%
%%Duration
Duration = normrnd(1,0.5,[NCustomers,1]);
for i=1:NCustomers
```

```
    if(Duration(i)<0)
        Duration(i)= 0;

    end
end
% xlswrite(filename,Duration,sheet,'AP3');
%%
%%
%%Discipline
Discipline = normrnd(7,2,[NCustomers,1]);
for i=1:NCustomers
    if(Discipline(i)<0)
        Discipline(i)= 0;
    elseif(Discipline(i)>10)
        Discipline(i)= 10;


    end
end
% xlswrite(filename,Discipline,sheet,'AQ3');
%%
%%Pillion
Pillion_Column = rand(NCustomers,1);
Pillion = zeros(NCustomers,2);
for i = 1:NCustomers
   if(Pillion_Column(i)<0.3)
      Pillion(i,1) = 1;
   elseif(Pillion_Column(i)>0.3)
       Pillion(i,2) = 1;
   end
end
% xlswrite(filename,Pillion,sheet,'AR3');
%%
%%RHabits
RHabits_Column = rand(NCustomers,1);
RHabits = zeros(NCustomers,3);
for i = 1:NCustomers
   if(RHabits_Column(i)<0.2)
      RHabits(i,1) = 1;
   elseif(RHabits_Column(i)>0.2 && RHabits_Column(i)<0.6)
       RHabits(i,2) = 1;
   elseif(RHabits_Column(i)>0.6 && RHabits_Column(i)<1)
       RHabits(i,3) = 1;
   end
end

% xlswrite(filename,RHabits,sheet,'AT3');
```

```
%%
%%
%%Complaint
Complaint = normrnd(7,2,[NCustomers,1]);
for i=1:NCustomers
    if(Complaint(i)<0)
        Complaint(i)= 0;
    elseif(Complaint(i)>10)
        Complaint(i)= 10;
    end
end
% xlswrite(filename,Complaint,sheet,'AW3');
%%
%%Outlook
Outlook_Column = rand(NCustomers,1);
Outlook = zeros(NCustomers,2);
for i = 1:NCustomers
   if(Outlook_Column(i)<0.7)
      Outlook(i,1) = 1;
   elseif(Outlook_Column(i)>0.3)
       Outlook(i,2) = 1;
   end
end
% xlswrite(filename,Outlook,sheet,'AX3');
%%
%%Income
Income_Column = rand(NCustomers,1);
Income = zeros(NCustomers,3);
for i = 1:NCustomers
   if(Income_Column(i)<0.6)
      Income(i,1) = 1;
   elseif(Income_Column(i)>0.6 && Income_Column(i)<0.95)
       Income(i,2) = 1;
   elseif(Income_Column(i)>0.95 && Income_Column(i)<1)
       Income(i,3) = 1;
   end
end
% xlswrite(filename,Income,sheet,'AZ3');

%%
%%Religion
Religion_Column = rand(NCustomers,1);
Religion = zeros(NCustomers,4);
for i = 1:NCustomers
   if(Religion_Column(i)<0.3)
      Religion(i,1) = 1;
   elseif(Religion_Column(i)>0.3 && Religion_Column(i)<0.4)
```

```
        Religion(i,2) = 1;
    elseif(Religion_Column(i)>0.4 && Religion_Column(i)<0.8)
        Religion(i,3) = 1;
    elseif(Religion_Column(i)>0.8 && Religion_Column(i)<1)
        Religion(i,4) = 1;
    end
end

% xlswrite(filename,Religion,sheet,'BC3');

%%
%%House
House_Column = rand(NCustomers,1);
House = zeros(NCustomers,2);
for i = 1:NCustomers
    if(House_Column(i)<0.6)
       House(i,1) = 1;
    elseif(House_Column(i)>0.6)
        House(i,2) = 1;
    end
end
% xlswrite(filename,House,sheet,'BG3');
%%
%%House
% House_Column = rand(NCustomers,1);
% House = zeros(NCustomers,3);
% for i = 1:NCustomers
%    if(House_Column(i)<0.6)
%       House(i,1) = 1;
%    elseif(House_Column(i)>0.6)
%        House(i,2) = 1;
%    end
% end
% xlswrite(filename,House,sheet,'BG3');

%%
%%Cars
Cars_Column = rand(NCustomers,1);
Cars = zeros(NCustomers,3);
for i = 1:NCustomers
    if(Cars_Column(i)<0.6)
       Cars(i,1) = 1;
    elseif(Cars_Column(i)>0.6 && Cars_Column(i)<0.8)
        Cars(i,2) = 1;
    elseif(Cars_Column(i)>0.8 && Cars_Column(i)<1)
        Cars(i,3) = 1;
    end
```

```
end

% xlswrite(filename,Cars,sheet,'BI3');
%%
Location_Section_Column = rand(NCustomers,1);
Location_Section = zeros(NCustomers,5);
for i = 1:NCustomers
   if(Location_Section_Column(i)<0.25)
       Location_Section(i,1) = 1;
   elseif(Location_Section_Column(i)>0.25 &&
Location_Section_Column(i)<0.35)
       Location_Section(i,2) = 1;
   elseif(Location_Section_Column(i)>0.35 &&
Location_Section_Column(i)<0.45)
       Location_Section(i,3) = 1;
   elseif(Location_Section_Column(i)>0.45 &&
Location_Section_Column(i)<0.75)
       Location_Section(i,4) = 1;
   elseif(Location_Section_Column(i)>0.75 && Location_Section_Column(i)<1)
       Location_Section(i,5) = 1;
   end
end
% xlswrite(filename,Location_Section,sheet,'BL3');

%%
%%Bikes
Bikes_Column = rand(NCustomers,1);
Bikes = zeros(NCustomers,2);
for i = 1:NCustomers
   if(Bikes_Column(i)<0.6)
      Bikes(i,1) = 1;
   elseif(Bikes_Column(i)>0.6)
       Bikes(i,2) = 1;
   end
end
% xlswrite(filename,Bikes,sheet,'BQ3');
%%
%%Family
Family_Column = rand(NCustomers,1);
Family = zeros(NCustomers,3);
for i = 1:NCustomers
   if(Family_Column(i)<0.3)
      Family(i,1) = 1;
   elseif(Family_Column(i)>0.3 && Family_Column(i)<0.6)
       Family(i,2) = 1;
   elseif(Family_Column(i)>0.6 && Family_Column(i)<1)
       Family(i,3) = 1;
```

```
        end
    end
    % xlswrite(filename,Family,sheet,'BS3');

    %% Generating Failure Possibility Score
    FailureScoreMatrix = zeros(NCustomers,5);
    %Column 1- Experience
    %Column 2- Usage
    %Column 3- Terrain
    %Column 4- Maintanance Habit
    %Column 5- Riding Discipline
    %Scale of 5 used for all five columns, 5 being highest failure...
    %possibility
    for i=1:NCustomers
        % For Experience in years
        if(Experience(i)<=0.5)
            FailureScoreMatrix(i,1) = 5;
        elseif(Experience(i)<=1.5 && Experience(i)>0.5)
            FailureScoreMatrix(i,1) = 4;
        elseif(Experience(i)<=2.5 && Experience(i)>1.5)
            FailureScoreMatrix(i,1) = 3;
        elseif(Experience(i)<=3.5 && Experience(i)>2.5)
            FailureScoreMatrix(i,1) = 2;
        elseif(Experience(i)>3.5)
            FailureScoreMatrix(i,1) = 1;
        end
        %For Usage(Distance in km/day)
        if(Distance(i)<=3)
            FailureScoreMatrix(i,2) = 1;
        elseif(Distance(i)<=5 && Distance(i)>3)
            FailureScoreMatrix(i,2) = 2;
        elseif(Distance(i)<=8 && Distance(i)>5)
            FailureScoreMatrix(i,2) = 3;
        elseif(Distance(i)<=11 && Distance(i)>8)
            FailureScoreMatrix(i,2) = 4;
        elseif(Distance(i)>11)
            FailureScoreMatrix(i,2) = 5;
        end

        %For Terrain
        if(Location(i,1)==1)
            FailureScoreMatrix(i,3) = 1;
        elseif(Location(i,2)==1)
            FailureScoreMatrix(i,3) = 3;
        elseif(Location(i,3)==1)
            FailureScoreMatrix(i,3) = 5;
        end
```

```

    %For Maintenance Habits
    if(MHabits(i,1)==1)
        FailureScoreMatrix(i,4) = 1;
    elseif(MHabits(i,2)==1)
        FailureScoreMatrix(i,4) = 5;
    end

    %For Riding Discipline
%     FailureScoreMatrix(i,5) = round(Discipline(i)/2,0);
    FailureScoreMatrix(i,5) = round(Discipline(i)/2);

end

% Categorising into A,B,C,D
A = zeros(NCustomers,1);
B = zeros(NCustomers,1);
C = zeros(NCustomers,1);
D = zeros(NCustomers,1);
Category1=A;

SumFailureScoreMatrix = sum(FailureScoreMatrix,2);
for i = 1:NCustomers
    if(SumFailureScoreMatrix(i)<=7)
        A(i) = 1;
        Category1(i) = 1;
    elseif(SumFailureScoreMatrix(i)<=13 && SumFailureScoreMatrix(i)>7)
        B(i) = 1;
        Category1(i) = 2;
    elseif(SumFailureScoreMatrix(i)<=18 && SumFailureScoreMatrix(i)>13)
        Category1(i) = 3;
    elseif(SumFailureScoreMatrix(i)<=25 && SumFailureScoreMatrix(i)>18)
        Category1(i) = 4;
    end
end
% Category1 = Category1';

% xlswrite(filename,Category1,sheet,'BV3');

%% Generating Probability Matrix (Each column represents a type of failure
%and its value is the corresponding probability
Pa = [0 0.01 0.05 0.13  0.23 0.37 0.69];
Pb = [0 0.018 0.048 0.14 0.24 0.36 0.67 1];
Pc = [0 0.012 0.051 0.12 0.22 0.35 0.70 1];
Pd = [0 0.015 0.052 0.125 0.24 0.36 0.69 1];
RandomCol = zeros(NCustomers,1);
NMonths = 24;
```

```
Failure = zeros(NCustomers,NMonths);
% for i = 1:NCustomers
%     Generating a row of random numbers, each belonging to a month
%     RandomCol = rand(1,NMonths);
%     for j = 1:NMonths
%        if(A(i))
%            if(RandomCol(j)<0.35)  %Failure prob = 10%
%                Failure(i,j) = 1;
%            end
%        elseif(B(i))
%            if(RandomCol(j)<0.38) %Failure prob = 12%
%                Failure(i,j) = 1;
%            end
%        elseif(C(i))
%            if(RandomCol(j)<0.42) %Failure prob = 14%
%                Failure(i,j) = 1;
%            end
%        elseif(D(i))
%            if(RandomCol(j)<0.47) %Failure prob = 18%
%                Failure(i,j) = 1;
%            end
%        end
%     end
% end

%% asdf

beta_A = 2;
beta_B = 2;
beta_C = 2;
beta_D = 2;
eta_A = 100;
eta_B = 80;
eta_C = 50;
eta_D = 30;
for i = 1:NCustomers
    Time = 0;
    if(A(i))
        while(Time <= NMonths  )
            mttf = wblrnd(eta_A,beta_A);
            Time = Time + mttf/30;
            Time = round(Time);
            if(Time <24 && Time > 0 )
            Failure(i,Time) = 1;
            end
        end
    end
```

```
     if(B(i))
        while(Time <= 24  )
            mttf = wblrnd(eta_B,beta_B);
            Time = Time + mttf/30;
            Time = round(Time);
            if(Time <24 && Time > 0)
            Failure(i,Time) = 1;
            end
        end
     end
     if(C(i))
        while(Time <= 24 )
            mttf = wblrnd(eta_C,beta_C);
            Time = Time + mttf/30;
            Time = round(Time);
            if(Time <24 && Time > 0)
            Failure(i,Time) = 1;
            end
        end
     end
    if(D(i))
        while(Time <= 24  )
            mttf = wblrnd(eta_D,beta_D);
            Time = Time + mttf/30;
            Time = round(Time);
            if(Time <24 && Time > 0)
            Failure(i,Time) = 1;
            end
        end
    end
end




%% asdfa

for i=1:NCustomers
   for j=1:NMonths
       if(Failure(i,j)==1)
           Ftype = rand;
           if(A(i))
               for k=1:size(Pa,2)-1
                   if(Ftype>Pa(1,k) && Ftype<=Pa(1,k+1))
                       Failure(i,j) = k;
                   end
               end
```

```
            end
            if(B(i))
                for k=1:size(Pb,2)-1
                    if(Ftype>Pb(1,k) && Ftype<=Pb(1,k+1))
                        Failure(i,j) = k;
                    end
                end
            end
            if(C(i))
                for k=1:size(Pc,2)-1
                    if(Ftype>Pc(1,k) && Ftype<=Pc(1,k+1))
                        Failure(i,j) = k;
                    end
                end
            end
            if(D(i))
                for k=1:size(Pd,2)-1
                    if(Ftype>Pd(1,k) && Ftype<=Pd(1,k+1))
                        Failure(i,j) = k;
                    end
                end
            end
        end
    end
end
% xlswrite(filename,Failure,sheet,'BW3');

%% Checking frequency of each failure
FailureCount1 = 0;
FailureCount2 = 0;
FailureCount3 = 0;
FailureCount4 = 0;
FailureCount5 = 0;
FailureCount6 = 0;
FailureCount7 = 0;

for i=1:NCustomers
    for j=1:24
        if(Failure(i,j)==1)
            FailureCount1 = FailureCount1 +1;
        elseif(Failure(i,j)==2)
            FailureCount2 = FailureCount2 +1;
        elseif(Failure(i,j)==3)
            FailureCount3 = FailureCount3 +1;
        elseif(Failure(i,j)==4)
            FailureCount4 = FailureCount4 +1;
        elseif(Failure(i,j)==5)
```

```
            FailureCount5 = FailureCount5 +1;
        elseif(Failure(i,j)==6)
            FailureCount6 = FailureCount6 +1;
        elseif(Failure(i,j)==7)
            FailureCount7 = FailureCount7 +1;
        end
    end
end

%% Money spent on each failure
%Money spent by Customer for each failure, each column represents cost for
%corresponding failure number

%CustomerMoney = [CM1 CM2 CM3 CM4 CM5 CM6 CM7];
CustomerMoney = [7000 300 500 400 300 200 100]; %Placing random values just
fro example

%Money spent by Service Provider for each failure, each column represents
cost for
%corresponding failure number
ServiceMoney = [-4000 -3400 -2000 -1000 -500 -400 -200]; %Placing random
values just for example

%Money spent by Manufacturer for each failure, each column represents cost
for
%corresponding failure number
%ManufacturerMoney = [MM1 MM2 MM3 MM4 MM5 MM6 MM7];
ManufacturerMoney = [7000 300 3500 700 900 1200 200]; %Placing random
values just for example

%% EMI Cash flow for customers
% Assuming 20% people take bike on EMI

RandRow = rand(NCustomers,1);
Loan = RandRow<=0.2;

% Assuming EMI = 7350 for bike price 1,50,000 on a ROI = 8.75% for 24
months
EMI = 7350;
EMICash = Loan*EMI;
EMICashFlow=repmat(EMICash,1,24);
%
% for i=1:24
%     EMICashFlow(:,i) = EMICash(:,1);
% end

%% Cost Matrix creation
```

```
CustomerCostMatrix = zeros(NCustomers,24) + EMICashFlow; %Adding EMI to the
Cash flow of Cost to Customer
ManufacturerCostMatrix = zeros(NCustomers,24) - EMICashFlow; %subtracting
EMI to the Cash flow of Cost to Manufacturer
ServiceProviderCostMatrix = zeros(NCustomers,24);

for i = 1:NCustomers
    for j = 1:24
        if Failure(i,j) ~= 0
        CustomerCostMatrix(i,j) = CustomerMoney(1,Failure(i,j));
        ManufacturerCostMatrix(i,j) = ManufacturerMoney(1,Failure(i,j));
        ServiceProviderCostMatrix(i,j) = ServiceMoney(1,Failure(i,j));
        end
    end
end
xlswrite('Cost_Matrix.xlsx',CustomerCostMatrix,1,'B2');
xlswrite('Cost_Matrix.xlsx',ServiceProviderCostMatrix,2,'B2');
xlswrite('Cost_Matrix.xlsx',ManufacturerCostMatrix,3,'B2');



%% Finding Time value of money for sevice provider and manufacturer
%rate of interest is 'rate' %
% rate = 0.04;
% ServiceProviderTotal = 0;
% ManufacturerTotal = 0;
% for i = 1:NCustomers
%     for j = 1:NMonths
%         ServiceProviderTotal = ServiceProviderTotal +
ServiceProviderCostMatrix(i,j)*(1+rate)^(NMonths-j);
%         ManufacturerTotal = ManufacturerTotal +
ManufacturerCostMatrix(i,j)*(1+rate)^(NMonths-j);
%     end
% end
% CPV = [];
% CPV(1:NCustomers,1:15) = rand(NCustomers,15);
% sumMatrix = sum(CPV,2);
% for i=1:size(CPV,1)
%     CPV(i,:) = CPV(i,:)/sumMatrix(i);
% end
% %xlswrite(filename,CPV,10,'B2');

%---- writing to datafile
dataToWrite=cat(2,age,Gender,Marital_status,Position_in_family,Occupation,E
ducation,Affiliation,Location,Purpose,Experience,Weight,Height,MHabits,Attr
action,Distance,Duration,Discipline,Pillion,RHabits,Complaint,Outlook,Incom
e,Religion,House,Cars,Location_Section,Bikes,Family,Category1,Failure);
```

```
xlswrite(filename,dataToWrite,sheet,'C3');
toc
```

Appendix II

Shiny App Code

```
library(shiny)
library(readxl)
library(shinyjs)
library(shinythemes)
library(ggplot2)
library(shinydashboard)
library(corrplot)
library(tableHTML)
library(DT)
library(randomForest)
library(dplyr)
library(caret)
library(e1071)
library(lattice)
library(rintrojs)
library(ggthemes)

ui <-
  dashboardPage(skin = "black",
                dashboardHeader(title = "Dual Degree Project",titleWidth =
300),
                dashboardSidebar(
                  sidebarMenu(
                    menuItem("Home", tabName = "Home", icon =
icon("home")),
                    menuItem("Upload Data here!", tabName = "Data", icon =
icon("table")),
                    menuItem("Predictions", tabName = "LR", icon =
icon("line-chart")),
                    menuItem("Data Insight", tabName = "DI", icon =
icon("eye")),
                    menuItem("Contact", tabName = "About", icon =
icon("address-book"))
                  )
                ),

                dashboardBody(

                  tags$head(
                    tags$style(HTML("
                                    @import
url('//fonts.googleapis.com/css?family=Georgia|Cabin:400,700');
```

```
                                     h1 {
                                     font-family: 'Georgia';
                                     font-weight: normal;
                                     line-height: 1;
                                     color: black;
                                     }
                                     h5 {
                                     font-family: 'Georgia';
                                     font-weight: normal;
                                     line-height: 1.1;
                                     color: black;
                                     font-size = 24px;
                                     font-variant: small-caps;
                                     }

                                     h4 {
                                     font-family: 'Georgia';
                                     font-weight: bold;
                                     font-color = red;
                                     line-height: 1;
                                     color: black;
                                     font-size = 20px;
                                     }

                                     "))
                     ),


                   fluidPage(
                     tags$style(make_css(list('.box',
                                              c('font-size', 'font-family',
'color'),
                                              c('14px', 'Georgia',
'Grey')))),

                     tags$head(tags$style(HTML('
                                               .main-header .logo {
                                               font-family: "Georgia",
Times, "Georgia", serif;
                                               font-weight: bold;
                                               font-size: 24px;
                                               }
                                               '))),

                     #Selecting theme
                     #shinythemes::themeSelector(),
```

```
                    #theme = shinytheme("united"),
                    useShinyjs(),
                    fluidRow(
                      img(height = 100,
                          width =
100,src="https://upload.wikimedia.org/wikipedia/en/thumb/5/58/IIT_Bombay_Lo
go.svg/1200px-IIT_Bombay_Logo.svg.png",
                          align = "left"),
                      img(height = 100,
                          width =
100,src="https://i.pinimg.com/originals/eb/0e/d5/eb0ed51dac78c6f5873bcb8099
416401.png",
                          align = "right"),


                      column(8,h1("Data Analytics in Warranty
Management"),align = "center",offset = 1),


                      HTML('<hr style="color: white;">')

                    ),
                    ##Making tabs
                    tabItems(
                      #Tab1 - Home
                      tabItem(tabName = "Home",icon = icon("home"),
                              #sidebarLayout(
                              # sidebarPanel( titlePanel(h3("Application of
Data
                              #                             Analytics in
Warranty Management"))),
                              #
                              #mainPanel (h6("This application is focused
on analyzing data related to royal
                              #             Enfield customers.
                              #            Machine learning and cash flow
analysis are incorporated to
                              #           optimize warranty policy.
                              #          ")
                              #                     )
                              #                    ),
                              box(width = 20,height = 5),
                              fluidRow(
                                infoBox(icon = icon("bullseye","fa-
1.5x"),title = h5("Aim"),value = h4("Application of Data Analytics in
Warranty Management"),
```

```
                                        width = "100%",fill = FALSE,color =
"green")
                              ),
                              fluidRow(
                                infoBox(icon = icon("angle-double-up","fa-
1.5x"),title = h5("Purpose"),
                                        value = h4("This application is
focused on analyzing data related to Royal
                                                   Enfield customers.
Machine learning and cash flow analysis are incorporated to
                                                   optimize warranty
policy."),
                                        width = "100%",fill = FALSE,color =
"purple")
                                ),
                              fluidRow(
                                infoBox(icon = icon("check","fa-1.5x"
),title = h5("Deliverables"),
                                        value = h4("A. B. C. D."),
                                        width = "100%",fill = FALSE,color =
"light-blue")
                              ),
                              fluidRow(
                                infoBox(icon = icon("file","fa-1.5x"),title
= h5("Project Report"),
                                        value = h4("Click to view
report"),href = "https://bighome.iitb.ac.in/index.php/s/A58PBrp8NnmkiWJ",
                                        width = "100%",fill = FALSE,color =
"aqua")
                              )

                                ),  #End of Tab1
                      #Tab 2 - Load Data
                      tabItem(tabName = "Data",icon = icon("table"),
                              tabsetPanel(
                                tabPanel("Select Data",
                                         box(
                                           fileInput("file1", 'Choose
Failure Data and Customer profile CSV File',
                                                     accept=c('text/csv',
'text/comma-separated- values,text/plain', '.csv'))
                                         ,width = 6),
                                         box(
                                           fileInput("file2", 'Choose
Ratings CSV File',
                                                     accept=c('text/csv',
'text/comma-separated- values,text/plain', '.csv'))
```

```
                                         ,width = 6)
                                ),
                                tabPanel("View Data Tables & Summary",
                                         "Summary",
                                         DTOutput(outputId =
'DataSummary'),

                                         fluidRow(
                                           actionButton("hideshow",
"Show/Hide Data"),
                                           div(style = 'overflow-x:
scroll', DT::dataTableOutput('tableOutput'))
                                           #DTOutput(outputId =
'tableOutput')
                                         ),
                                         fluidRow(
                                           actionButton("hideshow3",
"Show/Hide Data"),
                                           div(style = 'overflow-x:
scroll', DT::dataTableOutput('tableOutputCPV'))
                                         )
                                )
                              )
                              #checkboxInput("showModel1", "Show/Hide Model
1", value = FALSE)

                      ), #End of Tab 2
                      #Tab 3
                      tabItem(tabName = "LR",icon = icon("line-chart", lib
= "font-awesome"),
                              tabsetPanel(
                                tabPanel("Linear Regression",
                                         box(
                                           selectInput('xcol', 'X
Variable', "abc", selected = "Please select data first"),
                                           selectInput('ycol', 'Y
Variable', "pqr", selected = ""),
                                           width = "100%"
                                         ),
                                         box(
                                           plotOutput("regression"),
                                           width = "100%"
                                           #, plotOutput("linear1")
                                         )

                                ),
                                tabPanel("Algos",
```

```
                                          box(
                                            selectInput('algo_name',
                                                        'Please select an
algorithm',
                                                        "Please select data
first"), width = 6),
                                          box(
                                            selectInput('y_var',
                                                        'Please select month
to predict failure',
                                                        "Please select data
first"),
                                            width = 6),
                                          downloadButton("downloadData",
"Download"),
                                          verbatimTextOutput('conf_matrix')

#fluidRow(column(7,dataTableOutput('dto')))
                                          #tableOutput('conf_matrix_csv')

                                 )
                               )
                      ), #End of tab 3
                      tabItem(tabName = "DI",icon = icon("eye"),
                              tabsetPanel(

                                # tabPanel("Show/Hide Data",
                                #          fluidRow(
                                #           actionButton("hideshow2",
"Show/Hide Data")
                                #
                                #       )
                                # ),

                                tabPanel("Random Forest","Importance Plot",
                                         fluidRow(
                                           plotOutput("rfplot")
                                         )
                                ),
                                tabPanel("Data Insights",

                                         fluidRow(
                                           box(
                                             selectInput('feature1',
'Select 1st Feature', "abc"),
                                             selectInput('feature2',
'Select 2nd Feature', "pqr", selected = "")
```

```
                                            ),
                                            box(selectInput('A', 'Select 1st
Factor', "abc"),
                                                selectInput('B', 'Select 2nd
Factor', "pqr", selected = ""),
                                                selectInput('C', 'Select 3rd
Factor', "abc")
                                            ),

box(DTOutput('tableOutput2'),width =
                                                  "100%"),

                                            plotOutput(outputId =
'plot1',width = "100%",height = 600)

                                            # plotOutput(outputId =
'corrplot')
                                            #actionButton("hideshow2",
"Show/Hide Data"),
                                            #tableOutput(outputId =
'tableOutput2')
                                          )
                                 ),

                                 tabPanel("Data Summary",
                                          "Summary",
                                          tableOutput(outputId =
'DataSummary2')
                                 )

                               )
                               #checkboxInput("showModel1", "Show/Hide Model
1", value = FALSE)

                       ),

                       tabItem(tabName = "About", icon = icon("address-
book"),



                               box(
                                 tags$div(class = "header", checked = NA,
                                          tags$h4("Guided by- Prof. A.
Subash Babu"),
                                          tags$img(height = 200,
```

```
                                                  width =
200,src="http://www.akgec.in/sites/default/files/styles/testimonial_70x70/p
ublic/a_subash_babu.jpg?itok=tQb4yWby",
                                                  align = "left")
                                ),
                                actionButton(inputId='homepage',
label="Homepage",
                                             icon = icon("home"),
                                             onclick
="window.open('http://www.me.iitb.ac.in/faculty/48/profile/', '_blank')"
                                )
                              ),
                              box(
                                tags$div(class = "header", checked = NA,
                                         tags$h4("Created by- Mr. Yash A.
Baley"),
                                         tags$img(height = 200,
                                                  width =
200,src="https://media.licdn.com/dms/image/C5103AQGiykSKccxcJQ/profile-
displayphoto-
shrink_200_200/0?e=1530270000&v=beta&t=oTtotYG1zm2yB_3YJ_mWa2jvHHRADEjmebbI
JIxUTFQ",
                                                  align = "left")
                                ),
                                actionButton(inputId='linkedin',
label="LinkedIn",
                                             icon = icon("linkedin"),
                                             onclick
="window.open('https://www.linkedin.com/in/yash-a-baley-52301281/',
'_blank')"
                                ),
                                tags$br(),
                                actionButton(inputId='Facebook',
label="Facebook",
                                             icon = icon("facebook"),
                                             onclick
="window.open('https://www.facebook.com/YashABaley', '_blank')"
                                )
                              )



                      )
                              )
                    )

                  )
```

```
                  )
#End of Ui


server <- function(input,output,session){

  runjs('
        var el2 = document.querySelector(".skin-black");
        el2.className = "skin-black sidebar-mini";
        ')

  myData <- reactive({
    inFile <- input$file1
    if (is.null(inFile))
      return(NULL)

    tbl <- read.csv(inFile$datapath, header = TRUE)



    failure <- tbl[,(ncol(tbl)-23):ncol(tbl)]
    abc <- ifelse(failure>1,1,0)
    TotFailure <- rowSums(abc)
    tbl["Total.Failures"] <- TotFailure

    #Algo_names = data.frame(c("Linear Regression", "Logistic Regression",
"KNN", "SVM", "Random Forest"))

    updateSelectInput(session, inputId = 'xcol', label = 'X Variable',
                      choices = names(tbl), selected = names(tbl)[35])
    updateSelectInput(session, inputId = 'ycol', label = 'Y Variable',
                      choices = names(tbl), selected =
names(tbl)[ncol(tbl)])

    updateSelectInput(session, inputId = 'algo_name', label = 'Please
select an algorithm',
                      choices = c("Linear Regression", "Logistic
Regression", "KNN", "SVM", "Random Forest"),
                      selected = "SVM")

    updateSelectInput(session, inputId = 'y_var', label = 'Please select
month to predict failure',
                      choices = c("Total failures",1:24),
                      selected = 1)


    return(tbl)
```

```r

  })

  output$tableOutput <- renderDT({
    myData()
  })

  output$DataSummary <- renderDT({
    summary(myData())
  })

  results_table <- reactive({
    inFile <- input$file1
    if (is.null(inFile))
      return(NULL)

    tbl <- read.csv(inFile$datapath, header = TRUE)

    df3 <- data.frame(matrix(c(tbl[,"Km.day"],tbl[,"number.of.years"],

tbl[,"City"],tbl[,"Village"],tbl[,"Mountain"],

tbl[,"Regular"],tbl[,"Occasional"],tbl[,"Scale.of.10"]),
                             nrow = nrow(tbl), ncol = 8))
    ptm <- proc.time()
    x = df3
    month = as.numeric(input$y_var)
    y <- tbl[,74+month]
    y = ifelse(y>0,1,0)
    y = as.factor(y)
    x = scale(x, center = TRUE, scale = TRUE)

    inTrain = createDataPartition(y, p = 0.8,list = FALSE)
    NCustomers_train = 0.8*nrow(x)
    Train = x[1:NCustomers_train,]
    Test = x[NCustomers_train:nrow(x),]
    Trainy = y[1:NCustomers_train]
    Testy = y[NCustomers_train:nrow(x)]

    dataframe <- data.frame(x,y)
    traindata <- data.frame(Train,Trainy)
    as.data.frame(traindata)

    if(input$algo_name == "KNN") {

      model = train(x = Train, y= Trainy, method = "knn" )
      pred = predict(model, Test)
```

```
      result_matrix = confusionMatrix(pred, Testy)
      time_elapsed = proc.time() - ptm
      return(result_matrix)
    }

    if(input$algo_name == "SVM") {
      model_svm = svm(Train, Trainy)
      pred_svm = predict(model_svm,Test)
      result_matrix = confusionMatrix(pred_svm, Testy)
      return(result_matrix)
    }

    if(input$algo_name == "Random Forest") {
      traindata <- data.frame(Train,Trainy)
      testdata <- data.frame(Test,Testy)
      rf = randomForest(traindata$Trainy ~., ntree = 1000, data =
traindata)
      pred_rf = predict(rf , Test)

      result_matrix = confusionMatrix(pred_rf, Testy)
      return(result_matrix)
    }

    if(input$algo_name == "Logistic Regression") {

      traindata <- data.frame(Train,Trainy)
      testdata <- data.frame(Test,Testy)
      lr = glm(traindata$Trainy ~., data = traindata,
               family = binomial(link="logit"))
      pred_lr = round(predict(lr , testdata,type = "response"))
      pred_lr[1] = 1
      pred_lr = factor(pred_lr)
      result_matrix = confusionMatrix(pred_lr, Testy)
      return(result_matrix)
    }


    else {
      statement <- "Please select an Algo"
      return(statement)
    }

  })
  output$conf_matrix <- renderPrint({
    results_table()
  })

```

```
  conf_matrix_csv <- reactive({
    ab <- results_table()
    cd <- as.data.frame.matrix(ab)
    bc <- data.frame(matrix(unlist(ab), nrow=12,
byrow=T),stringsAsFactors=FALSE)
    #ab <-
data.frame(cbind(t(results_table()$overall),t(results_table()$byClass)))
    return(cd)
  })



  output$dto <- renderDataTable(conf_matrix_csv(), extensions = 'Buttons',
                                options = list(dom = 'Bfrtip',
                                               buttons = c('copy', 'csv',
'excel', 'pdf', 'print')))

  myData2CPV <- reactive({
    inFile2 <- input$file2
    if (is.null(inFile2))
      return(NULL)

    tbl2 <- read.csv(inFile2$datapath, header = TRUE)
    tbl2 <- tbl2[,2:17]

    #failure <- tbl[,(ncol(tbl)-23):ncol(tbl)]
    #TotFailure <- rowSums(failure)
    #tbl["Total.Failures"] <- TotFailure



    updateSelectInput(session, inputId = 'feature1', label = 'Select 1st
Feature',
                      choices = names(tbl2), selected = names(tbl2)[3])
    updateSelectInput(session, inputId = 'feature2', label = 'Select 2st
Feature',
                      choices = names(tbl2), selected = names(tbl2)[2])

    updateSelectInput(session, inputId = 'A', label = 'Select 1st factor',
                      choices = names(myData()), selected =
names(myData())[4])
    updateSelectInput(session, inputId = 'B', label = 'Select 2nd factor',
                      choices = names(myData()), selected =
names(myData())[5])
    updateSelectInput(session, inputId = 'C', label = 'Select 3rd factor',
                      choices = names(myData()), selected =
names(myData())[6])
```

```


      return(tbl2)

    })

    myData3<- reactive({
      CPV <- myData2CPV()[,ncol(myData2CPV())]
      feature1_rating <- myData2CPV()[,input$feature1]
      feature2_rating <- myData2CPV()[,input$feature2]

      first <- c(input$A,input$B,input$C)
      first <- as.matrix(first)
      a1 <- first[1,]
      b1 <- first[2,]
      c1 <- first[3,]
      mean_value = matrix(0,nrow = nrow(first),ncol = 4)
      for (i in 1:nrow(first)){
        mean_value[i,1] <- sum( myData()[,first[i,]]>0)
        mean_value[i,2] <- sum(myData()[,first[i,]]*CPV)/mean_value[i,1]
        mean_value[i,3] <-
  sum(myData()[,first[i,]]*feature1_rating)/mean_value[i,1]
        mean_value[i,4] <-
  sum(myData()[,first[i,]]*feature2_rating)/mean_value[i,1]
      }

      df1 <- data.frame(
        Rating_Type = factor(c(rep("Overall",nrow(first)),
                               rep(input$feature1,nrow(first)),
                               rep(input$feature2,nrow(first)))),
        time = factor(c(rep(first,nrow(first)))),
        levels=c(first)

      )

      df1$Mean_rating <-
  c(mean_value[1,2:4],mean_value[2,2:4],mean_value[3,2:4])
      return(df1)
    })

    output$plot1 <- renderPlot({
      ggplot(data=myData3(), aes(x=time, y=Mean_rating, group=Rating_Type,
                                 shape=Rating_Type,color = Rating_Type),
             environment = environment() )+
        geom_line(size = 1.5) +
        geom_point(size = 1.5) +
        labs(x="", y = "Ratings") +
```

```
      theme(axis.text=element_text(size=16,face = "bold"),
            axis.title=element_text(size=16,face="bold"),
            legend.text=element_text(size=16,face="bold"),
            legend.title=element_text(size=16,face="bold"),
            legend.key.size = unit(2,"line"))


  })

  output$rfplot <- renderPlot({
    tbl <- myData2CPV()[,2:ncol(myData2CPV())]
    rf_out <- randomForest(CPV ~ ., data=tbl)


    # Sorts by variable importance and relevels factors to match ordering
    var_importance <- data_frame(variable=setdiff(colnames(tbl), "CPV"),
                                 importance=as.vector(importance(rf_out)))
    var_importance <- arrange(var_importance, desc(importance))
    var_importance$variable <- factor(var_importance$variable,
levels=var_importance$variable)

    p <- ggplot(var_importance, aes(x=variable, weight=importance,
fill=variable))
    p <- p + geom_bar() + ggtitle("Variable Importance from Random Forest
Fit")
    p <- p + xlab("Parameters") + ylab("Variable Importance (Contribution
towards overall CPV)")
    p <- p + scale_fill_discrete(name="Parameter Names")
    p <- p + theme(axis.text.x=element_blank(),
              axis.text.y=element_text(size=12),
              axis.title=element_text(size=16),
              plot.title=element_text(size=18),
              legend.title=element_text(size=16),
              legend.text=element_text(size=12))
    p #+  geom_text(var_importance,aes(label=importance),
position=position_dodge(width=0.9), vjust=-0.25)


  })


  output$corrplot <- renderPlot({
    corrplot(as.matrix(myData2CPV()), is.corr = FALSE, method="square",
order="FPC", tl.srt = 90)
  })

  output$tableOutput2 <- renderDT({
```

```
    myData3()
  })

  output$tableOutputCPV <- renderDT({
    myData2CPV()
  })


  output$DataSummary2 <- renderTable({
    summary(myData2CPV())
  })

  output$regression <- renderPlot({
    ggplot(myData(),aes_string(x=input$xcol,y=input$ycol))  +
      geom_smooth(method='lm',formula=y~x)+ggtitle('Linear Regression
Curve')+
      theme(plot.title = element_text(color="black", size=16,
face="bold.italic"))

  })

  observeEvent(input$hideshow, {
    # every time the button is pressed, alternate between hiding and
showing the plot
    toggle("tableOutput")
  })

  observeEvent(input$hideshow3, {
    # every time the button is pressed, alternate between hiding and
showing the plot
    toggle("tableOutputCPV")
  })

  observeEvent(input$hideshow2, {
    # every time the button is pressed, alternate between hiding and
showing the plot
    toggle("tableOutput2")
  })


  output$linear1 <- renderPlot({
    #ggplot(myData(),aes_string(x=input$xcol,y=input$ycol))  +
geom_smooth(method='lm',formula=y~x)+ggtitle('Lm Curve')+theme(plot.title =
element_text(color="black", size=16, face="bold.italic"))
    #  plot(myData(),aes_string(x=input$xcol,y=input$ycol),ylim=c(0,20),
xlim=c(0,20))

```

```
  })

  output$downloadData <- downloadHandler(
    filename = function() {
      paste(input$algo_name,"_Month_",input$y_var ,".csv", sep = "")
    },
    content = function(file) {
      write.csv(conf_matrix_csv(), file )
    }
  )

}
shinyApp(ui = ui, server = server
```

Appendix III

R Code for CPV Optimization Algorithm

```
library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms
library(factoextra) # clustering algorithms & visualization
library(ggplot2)
library(ggfortify)
library(randomForest)
library(dplyr)

##################      Load data         ################
ratings <- read.csv("Ratings.csv")
cust <- read.csv('SimulatedData_400.csv')
df <- cust[,1:74]

################# Generating Data for LOS #################
NCustomers = 400
NFactors = 15
NLevels = 5
LOS = matrix(nrow = NCustomers,ncol = NFactors)
for (i in 1:NCustomers){
  LOS[i,] = sample(1:5,NFactors, replace = TRUE)
}
write.csv(LOS,"LOS.csv")

### Clustering ####
df1 <- data.frame(matrix(c(df[,"Age"],df[,"Unemployed"],

df[,"Entrepreneur"],df[,"Unskilled.Worker"],df[,"Skilled.Worker"],
                           df[,"Management"],df[,"Farmer"], df[,"City"],
                           df[,"Mountain"],df[,"Village"],
                           df[,"Kg"],df[,"Km.day"]),nrow = nrow(df), ncol =
12))

scaled.df1 <- scale(df1)

# check that we get mean of 0 and sd of 1
colMeans(scaled.df1)  # faster version of apply(scaled.dat, 2, mean)
apply(scaled.df1, 2, sd)

k1 <- kmeans(scaled.df1,centers = 4)

################# Seperating clusters #################
A <- df1[k1$cluster==1,]
```

```
B <- df1[k1$cluster==2,]
C <- df1[k1$cluster==3,]
D <- df1[k1$cluster==4,]

################ Variable importance #################
### For cluster A ###
set.seed(42)
ratings <- ratings[2:17]
ratings_A <- ratings[k1$cluster==1,]
rownames(ratings_A) <- 1:nrow(ratings_A)
rf_out_A <- randomForest(CPV ~ ., data=ratings_A)

# Extracts variable importance (Mean Decrease in Gini Index)
# Sorts by variable importance and relevels factors to match ordering
var_importance_A <- data_frame(variable=setdiff(colnames(ratings_A),
"CPV"),
                             importance=as.vector(importance(rf_out_A,scale
= FALSE)))
var_importance_A$serial <- c(1:nrow(var_importance_A))
var_importance_A <- arrange(var_importance_A, desc(importance))
var_importance_A$variable <- factor(var_importance_A$variable,
levels=var_importance_A$variable)

### For cluster B ###
set.seed(42)
ratings_B <- ratings[k1$cluster==2,]
rownames(ratings_B) <- 1:nrow(ratings_B)

rf_out_B <- randomForest(CPV ~ ., data=ratings_B)

# Extracts variable importance (Mean Decrease in Gini Index)
# Sorts by variable importance and relevels factors to match ordering
var_importance_B <- data_frame(variable=setdiff(colnames(ratings_B),
"CPV"),
                             importance=as.vector(importance(rf_out_B,scale
= FALSE)))
var_importance_B$serial <- c(1:nrow(var_importance_B))
var_importance_B <- arrange(var_importance_B, desc(importance))
var_importance_B$variable <- factor(var_importance_B$variable,
levels=var_importance_B$variable)

### For cluster C ###
set.seed(42)
ratings_C <- ratings[k1$cluster==3,]
rownames(ratings_C) <- 1:nrow(ratings_C)
rf_out_C <- randomForest(CPV ~ ., data=ratings_C)

```

```
# Extracts variable importance (Mean Decrease in Gini Index)
# Sorts by variable importance and relevels factors to match ordering
var_importance_C <- data_frame(variable=setdiff(colnames(ratings_C),
"CPV"),

importance=as.vector(importance(rf_out_C,scale = FALSE)))
var_importance_C$serial <- c(1:nrow(var_importance_C))
var_importance_C <- arrange(var_importance_C, desc(importance))
var_importance_C$variable <- factor(var_importance_Cvariable,
levels=var_importance_C$variable)


### For cluster D ###
set.seed(42)
ratings <- ratings[2:17]
ratings_D <- ratings[k1$cluster==4,]
rownames(ratings_D) <- 1:nrow(ratings_D)
rf_out_D <- randomForest(CPV ~ ., data=ratings_D)

# Extracts variable importance (Mean Decrease in Gini Index)
# Sorts by variable importance and relevels factors to match ordering
var_importance_D <- data_frame(variable=setdiff(colnames(ratings_D),
"CPV"),

importance=as.vector(importance(rf_out_D,scale = FALSE)))
var_importance_D$serial <- c(1:nrow(var_importance_D))
var_importance_D <- arrange(var_importance_D, desc(importance))
var_importance_D$variable <- factor(var_importance_D$variable,
levels=var_importance_D$variable)

############ Reading LOS file #############
LOS <- read.csv("LOS.csv")
LOS <- LOS[,-1]
### A type customers ###
LOS_A <- LOS[k1$cluster==1,]
rownames(LOS_A) <- 1:nrow(LOS_A)
NCustomers_A = nrow(LOS_A)
LOS_A <- data.frame(LOS_A)
imp_A <- as.vector(var_importance_A[,3])

LOS_A_Updated = LOS_A
for (i in 1:nrow(LOS_A)){
  for (j in imp_A ){
  if(LOS_A[i,j] < 3 && var_importance_A[var_importance_A$serial==j,2]>5 &&
var_importance_A[var_importance_A$serial==j,2]<=8){
    LOS_A_Updated[i,j] = 3
  }
```

```
    if(LOS_A[i,j] < 4 && var_importance_A[var_importance_A$serial==j,2]>8
&& var_importance_A[var_importance_A$serial==j,2]<=10){
      LOS_A_Updated[i,j] = 4
    }
    if(LOS_A[i,j] < 5 &&
var_importance_A[var_importance_A$serial==j,2]>10){
      LOS_A_Updated[i,j] = 5
    }
    if(LOS_A[i,j] > 2 && var_importance_A[var_importance_A$serial==j,2]>2.5
&& var_importance_A[var_importance_A$serial==j,2]<=5){
      LOS_A_Updated[i,j] = 2
    }
    if(LOS_A[i,j] > 1 &&
var_importance_A[var_importance_A$serial==j,2]<2.5){
      LOS_A_Updated[i,j] = 1
    }
  }
}


### B type customers ###
LOS_B <- LOS[k1$cluster==2,]
rownames(LOS_B) <- 1:nrow(LOS_B)
NCustomers_B = nrow(LOS_B)
LOS_B <- data.frame(LOS_B)
imp_B <- as.vector(var_importance_B[,3])

LOS_B_Updated = LOS_B
for (i in 1:nrow(LOS_B)){
  for (j in 1:NFactors ){
    if(LOS_B[i,j] < 3 && var_importance_B[var_importance_B$serial==j,2]>5
&& var_importance_B[var_importance_B$serial==j,2]<=8){
      LOS_B_Updated[i,j] = 3
    }
    if(LOS_B[i,j] < 4 && var_importance_B[var_importance_B$serial==j,2]>8
&& var_importance_B[var_importance_B$serial==j,2]<=10){
      LOS_B_Updated[i,j] = 4
    }
    if(LOS_B[i,j] < 5 &&
var_importance_B[var_importance_B$serial==j,2]>10){
      LOS_B_Updated[i,j] = 5
    }
    if(LOS_B[i,j] > 2 && var_importance_B[var_importance_B$serial==j,2]>2.5
&& var_importance_B[var_importance_B$serial==j,2]<=5){
      LOS_B_Updated[i,j] = 2
    }
```

```
    if(LOS_B[i,j] > 1 &&
var_importance_B[var_importance_B$serial==j,2]<2.5){
      LOS_B_Updated[i,j] = 1
    }
  }
}



### C type customers ###
LOS_C <- LOS[k1$cluster==3,]
rownames(LOS_C) <- 1:nrow(LOS_C)
NCustomers_C = nrow(LOS_C)
LOS_C <- data.frame(LOS_C)
imp_C <- as.vector(var_importance_C[,3])

LOS_C_Updated = LOS_C
for (i in 1:nrow(LOS_C)){
  for (j in 1:NFactors ){
    if(LOS_C[i,j] < 3 && var_importance_C[var_importance_C$serial==j,2]>5
&& var_importance_C[var_importance_C$serial==j,2]<=8){
      LOS_C_Updated[i,j] = 3
    }
    if(LOS_C[i,j] < 4 && var_importance_C[var_importance_C$serial==j,2]>8
&& var_importance_C[var_importance_C$serial==j,2]<=10){
      LOS_C_Updated[i,j] = 4
    }
    if(LOS_C[i,j] < 5 &&
var_importance_C[var_importance_C$serial==j,2]>10){
      LOS_C_Updated[i,j] = 5
    }
    if(LOS_C[i,j] > 2 && var_importance_C[var_importance_C$serial==j,2]>2.5
&& var_importance_C[var_importance_C$serial==j,2]<=5){
      LOS_C_Updated[i,j] = 2
    }
    if(LOS_C[i,j] > 1 &&
var_importance_C[var_importance_C$serial==j,2]<2.5){
      LOS_C_Updated[i,j] = 1
    }
  }
}

### D type customers ###
LOS_D <- LOS[k1$cluster==4,]
rownames(LOS_D) <- 1:nrow(LOS_D)
LOS_D <- data.frame(LOS_D)
NCustomers_D = nrow(LOS_D)
```

```
imp_D <- as.vector(var_importance_D[,3])

LOS_D_Updated = LOS_D
for (i in 1:nrow(LOS_D)){
  for (j in 1:NFactors ){
    if(LOS_D[i,j] < 3 && var_importance_D[var_importance_D$serial==j,2]>5
&& var_importance_D[var_importance_D$serial==j,2]<=8){
      LOS_D_Updated[i,j] = 3
    }
    if(LOS_D[i,j] < 4 && var_importance_D[var_importance_D$serial==j,2]>8
&& var_importance_D[var_importance_D$serial==j,2]<=10){
      LOS_D_Updated[i,j] = 4
    }
    if(LOS_D[i,j] < 5 &&
var_importance_D[var_importance_D$serial==j,2]>10){
      LOS_D_Updated[i,j] = 5
    }
    if(LOS_D[i,j] > 2 && var_importance_D[var_importance_D$serial==j,2]>2.5
&& var_importance_D[var_importance_D$serial==j,2]<=5){
      LOS_D_Updated[i,j] = 2
    }
    if(LOS_D[i,j] > 1 &&
var_importance_D[var_importance_D$serial==j,2]<2.5){
      LOS_D_Updated[i,j] = 1
    }

  }
}

############ Cost Matrix ##############
Cost_Matrix = matrix(nrow = NFactors, ncol = NLevels)
for (i in (1:NLevels)) {
  Cost_Matrix[,i] = runif(NFactors, min = 30+8*i,max = 35+8*i)
}

Cost_initial_A = 0
for (i in 1:NCustomers_A){
  for (j in 1:NFactors){
Cost_initial_A = Cost_initial_A + Cost_Matrix[j,LOS_A[i,j]]
  }
}


Cost_initial_B = 0
for (i in 1:NCustomers_B){
  for (j in 1:NFactors){
    Cost_initial_B = Cost_initial_B + Cost_Matrix[j,LOS_B[i,j]]
```

```
    }
  }
  Cost_initial_C = 0
  for (i in 1:NCustomers_C){
    for (j in 1:NFactors){
      Cost_initial_C = Cost_initial_C + Cost_Matrix[j,LOS_C[i,j]]
    }
  }
  Cost_initial_D = 0
  for (i in 1:NCustomers_D){
    for (j in 1:NFactors){
      Cost_initial_D = Cost_initial_D + Cost_Matrix[j,LOS_D[i,j]]
    }
  }
  Total_Initial_Cost = Cost_initial_D + Cost_initial_C + Cost_initial_B +
  Cost_initial_A

  ###### Updated costs ########
  Cost_Updated_A = 0
  for (i in 1:NCustomers_A){
    for (j in 1:NFactors){
      Cost_Updated_A = Cost_Updated_A + Cost_Matrix[j,LOS_A_Updated[i,j]]
    }
  }


  Cost_Updated_B = 0
  for (i in 1:NCustomers_B){
    for (j in 1:NFactors){
      Cost_Updated_B = Cost_Updated_B + Cost_Matrix[j,LOS_B_Updated[i,j]]
    }
  }
  Cost_Updated_C = 0
  for (i in 1:NCustomers_C){
    for (j in 1:NFactors){
      Cost_Updated_C = Cost_Updated_C + Cost_Matrix[j,LOS_C_Updated[i,j]]
    }
  }
  Cost_Updated_D = 0
  for (i in 1:NCustomers_D){
    for (j in 1:NFactors){
      Cost_Updated_D = Cost_Updated_D + Cost_Matrix[j,LOS_D_Updated[i,j]]
    }
  }

  Total_Updated_Cost = Cost_Updated_D + Cost_Updated_C + Cost_Updated_B +
  Cost_Updated_A
```

Appendix IV

R code for Number of Customer vs Accuracy

```r
library(caret)
library(gtable)
library(e1071)
library(randomForest)
library(lattice)
library(ggplot2)
library(gridExtra)
rm(list = ls())
acc <- matrix(nrow = 4,ncol = 30)
sens <- acc
spec <- acc

for (i in c(1:30)) {
  NCustomers_1 = 1000 + 100*i
  filename = 'SimulatedData_4000_eta_100.csv'
  df <- read.csv(filename)
  df <- df[1:(NCustomers_1),]
  df3 <- data.frame(matrix(c(df[,41],df[,34],
                             df[,28],df[,29],df[,30],
                             df[,37],df[,38],df[,43]),
                           nrow = nrow(df), ncol = 8))
  x = df3
  #month = as.numeric(input$y_var)
  month = 20
  NCustomers = nrow(df3)
  y <- df[,(74+month)]
  y = ifelse(y>0,1,0)
  y = as.factor(y)
  x = scale(x, center = TRUE, scale = TRUE)

  inTrain = createDataPartition(y, p = 0.8,list = FALSE)
  NCustomers_train = 0.8*nrow(x)
  Train = x[1:NCustomers_train,]
  Test = x[NCustomers_train:nrow(x),]
  Trainy = y[1:NCustomers_train]
  Testy = y[NCustomers_train:nrow(x)]

  dataframe <- data.frame(x,y)
  traindata <- data.frame(Train,Trainy)
  #as.data.frame(traindata)

  ##### Logistic #######
```

```

  traindata <- data.frame(Train,Trainy)
  testdata <- data.frame(Test,Testy)
  lr = glm(traindata$Trainy ~., data = traindata,
           family = binomial(link="logit"))
  pred_lr = round(predict(lr , testdata,type = "response"))
  pred_lr = factor(pred_lr)
  result_matrix = confusionMatrix(pred_lr, Testy)
  r_m <- as.table(result_matrix, what = "classess")
  acc[1,i] <- (r_m[1,1]+ r_m[2,2])/(r_m[1,1]+ r_m[1,2] + r_m[2,1]+
r_m[2,2])
  spec[1,i] <- specificity(r_m)
  sens[1,i] <- sensitivity(r_m)

  #Sensitivity = TP / TP + FN
  #Specificity = TN / TN + FP
  #Precision = TP / TP + FP

  #### KNN ######

  model = train(x = Train, y= Trainy, method = "knn" )
  pred = predict(model, Test)
  result_matrix = confusionMatrix(pred, Testy)
  r_m <- as.table(result_matrix, what = "classess")
  acc[2,i] <- (r_m[1,1]+ r_m[2,2])/(r_m[1,1]+ r_m[1,2] + r_m[2,1]+
r_m[2,2])
  spec[2,i] <- specificity(r_m)
  sens[2,i] <- sensitivity(r_m)


  ####### SVM ########
  model_svm = svm(Train, Trainy)
  pred_svm = predict(model_svm,Test)
  result_matrix = confusionMatrix(pred_svm, Testy)
  r_m <- as.table(result_matrix, what = "classess")
  acc[3,i] <- (r_m[1,1]+ r_m[2,2])/(r_m[1,1]+ r_m[1,2]+ r_m[2,1]+ r_m[2,2])
  spec[3,i] <- specificity(r_m)
  sens[3,i] <- sensitivity(r_m)


  ########## Random forest #######
  traindata <- data.frame(Train,Trainy)
  testdata <- data.frame(Test,Testy)
  rf = randomForest(traindata$Trainy ~., ntree = 1000, data = traindata)
  pred_rf = predict(rf , Test)
  result_matrix = confusionMatrix(pred_rf, Testy)
  r_m <- as.table(result_matrix, what = "classess")
```

```
  acc[4,i] <- (r_m[1,1]+ r_m[2,2])/(r_m[1,1]+ r_m[1,2]+ r_m[2,1]+ r_m[2,2])
  spec[4,i] <- specificity(r_m)
  sens[4,i] <- sensitivity(r_m)

}

write.csv(acc,'Accuracy1.csv')
write.csv(spec,'Spec1.csv')
write.csv(sens,'Sens1.csv')

TTF = 200 - 10*c(1:16)
Ncustomers_1 = 1000 + 100*c(1:30)

a1 <-
c(c("Logistic")[rep(1,30)],c("SVM")[rep(1,30)],c("KNN")[rep(1,30)],c("Rando
m Forest")[rep(1,30)])
b1 <- Ncustomers_1
c1 <- as.vector(acc)

df1 <- data.frame(algo_name = factor(a1,
                                     levels =
c("Logistic","SVM","KNN","Random Forest")),
                  NumberOfCustomers = rep(Ncustomers_1,4),
                  Accuracy = c1)
lp1 <- ggplot(data=df1, aes(x=NumberOfCustomers, y=Accuracy,
                            group=algo_name, shape=algo_name,
                            colour=algo_name)) + geom_smooth(se = F) +
geom_point()


df2 <- data.frame(algo_name = factor(a1,levels =
c("Logistic","SVM","KNN","Random Forest")),
                  NumberOfCustomers = rep(Ncustomers_1,4),
                  Sensitivity = as.vector(sens))
lp2 <- ggplot(data=df2, aes(x=NumberOfCustomers, y=Sensitivity,
                            group=algo_name, shape=algo_name,
                            colour=algo_name)) +
  geom_smooth(se = F) + geom_point()


df3 <- data.frame(algo_name = factor(a1,
                                     levels =
c("Logistic","SVM","KNN","Random Forest")),
                  NumberOfCustomers = rep(Ncustomers_1,4),
                  Specificity = as.vector(spec))

lp3 <- ggplot(data=df3, aes(x=NumberOfCustomers, y=Specificity,
```

```
                              group=algo_name, shape=algo_name,
                              colour=algo_name)) +
  geom_smooth(se = F) + geom_point()

grid.arrange(lp1,lp2,lp3,nrow = 2)
```

Appendix V

R Code for MTTF vs Accuracy

```
library(caret)
library(gtable)
library(e1071)
library(randomForest)
library(lattice)
library(ggplot2)
library(gridExtra)
rm(list=ls())
acc <- matrix(nrow = 4,ncol = 16)
sens <- acc
spec <- acc
for (i in c(1:16)) {
eta = 200 - i*10
filename = paste('SimulatedData_4000_eta_',as.character(eta),".csv",sep =
"")
df <- read.csv(filename)
df3 <- data.frame(matrix(c(df[,41],df[,34],
                           df[,28],df[,29],df[,30],
                           df[,37],df[,38],df[,43]),
                         nrow = nrow(df), ncol = 8))
x = df3
#month = as.numeric(input$y_var)
month = 20
NCustomers = nrow(df3)
y <- df[,(74+month)]
y = ifelse(y>0,1,0)
y = as.factor(y)
x = scale(x, center = TRUE, scale = TRUE)

inTrain = createDataPartition(y, p = 0.8,list = FALSE)
NCustomers_train = 0.8*nrow(x)
Train = x[1:NCustomers_train,]
Test = x[NCustomers_train:nrow(x),]
Trainy = y[1:NCustomers_train]
Testy = y[NCustomers_train:nrow(x)]

dataframe <- data.frame(x,y)
traindata <- data.frame(Train,Trainy)
#as.data.frame(traindata)

##### Logistic #######

```

```
traindata <- data.frame(Train,Trainy)
testdata <- data.frame(Test,Testy)
lr = glm(traindata$Trainy ~., data = traindata,
         family = binomial(link="logit"))
pred_lr = round(predict(lr , testdata,type = "response"))
pred_lr = factor(pred_lr)
result_matrix = confusionMatrix(pred_lr, Testy)
r_m <- as.table(result_matrix, what = "classess")
acc[1,i] <- (r_m[1,1]+ r_m[2,2])/(r_m[1,1]+ r_m[1,2] + r_m[2,1]+ r_m[2,2])
spec[1,i] <- specificity(r_m)
sens[1,i] <- sensitivity(r_m)

#Sensitivity = TP / TP + FN
#Specificity = TN / TN + FP
#Precision = TP / TP + FP

#### KNN ######

  model = train(x = Train, y= Trainy, method = "knn" )
  pred = predict(model, Test)
  result_matrix = confusionMatrix(pred, Testy)
  r_m <- as.table(result_matrix, what = "classess")
  acc[2,i] <- (r_m[1,1]+ r_m[2,2])/(r_m[1,1]+ r_m[1,2] + r_m[2,1]+
r_m[2,2])
  spec[2,i] <- specificity(r_m)
  sens[2,i] <- sensitivity(r_m)


####### SVM ########
  model_svm = svm(Train, Trainy)
  pred_svm = predict(model_svm,Test)
  result_matrix = confusionMatrix(pred_svm, Testy)
  r_m <- as.table(result_matrix, what = "classess")
  acc[3,i] <- (r_m[1,1]+ r_m[2,2])/(r_m[1,1]+ r_m[1,2]+ r_m[2,1]+ r_m[2,2])
  spec[3,i] <- specificity(r_m)
  sens[3,i] <- sensitivity(r_m)


########## Random forest #######
  traindata <- data.frame(Train,Trainy)
  testdata <- data.frame(Test,Testy)
  rf = randomForest(traindata$Trainy ~., ntree = 1000, data = traindata)
  pred_rf = predict(rf , Test)
  result_matrix = confusionMatrix(pred_rf, Testy)
  r_m <- as.table(result_matrix, what = "classess")
  acc[4,i] <- (r_m[1,1]+ r_m[2,2])/(r_m[1,1]+ r_m[1,2]+ r_m[2,1]+ r_m[2,2])
  spec[4,i] <- specificity(r_m)
```

```
  sens[4,i] <- sensitivity(r_m)

}

write.csv(acc,'Accuracy.csv')
write.csv(spec,'Spec.csv')
write.csv(sens,'Sens.csv')

TTF = 200 - 10*c(1:16)


a1 <-
c(c("Logistic")[rep(1,16)],c("SVM")[rep(1,16)],c("KNN")[rep(1,16)],c("Rando
m Forest")[rep(1,16)])
b1 <- TTF
c1 <- as.vector(acc)
Accuracy <- c1
df1 <- data.frame(algo_name = factor(a1,
               levels = c("Logistic","SVM","KNN","Random Forest")),
  TimeToFailure = rep(TTF,4),
  Accuracy = c1)
lp1 <- ggplot(data=df1, aes(x=TimeToFailure, y=Accuracy,
                            group=algo_name, shape=algo_name,
                            colour=algo_name)) + geom_smooth(se = F) +
geom_point()


df2 <- data.frame(algo_name = factor(a1,levels =
c("Logistic","SVM","KNN","Random Forest")),
                  TimeToFailure = rep(TTF,4),
                  Sensitivity = as.vector(sens))
lp2 <- ggplot(data=df2, aes(x=TimeToFailure, y=Sensitivity,
                            group=algo_name, shape=algo_name,
                            colour=algo_name)) +
                            geom_smooth(se = F) + geom_point()


df3 <- data.frame(algo_name = factor(a1,
                                     levels =
c("Logistic","SVM","KNN","Random Forest")),
                  TimeToFailure = rep(TTF,4),
                  Specificity = as.vector(spec))

lp3 <- ggplot(data=df3, aes(x=TimeToFailure, y=Specificity,
                            group=algo_name, shape=algo_name,
                            colour=algo_name)) +
  geom_smooth(se = F) + geom_point()
```

```
par(mfrow=c(3,1))
grid.arrange(lp1,lp2,lp3,nrow = 2)
```

Appendix VI

R Code for K-Means Clustering

```
library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms
library(factoextra) # clustering algorithms & visualization
library(ggplot2)
library(ggfortify)



df <- read.csv("Datafile_failures.csv")

df1 <- data.frame(matrix(c(df[,"Age"],df[,"Unemployed"],

df[,"Entrepreneur"],df[,"Unskilled.Worker"],df[,"Skilled.Worker"],
                           df[,"Management"],df[,"Farmer"], df[,"City"],
                           df[,"Mountain"],df[,"Village"],
                           df[,"Kg"],df[,"Km.day"]),nrow = nrow(df), ncol =
12))

scaled.df1 <- scale(df1)

# check that we get mean of 0 and sd of 1
colMeans(scaled.df1)  # faster version of apply(scaled.dat, 2, mean)
apply(scaled.df1, 2, sd)

k1 <- kmeans(df1,centers = 4)


df2 = df1[1:1000,]

scaled.df2 <- scale(df2)

# check that we get mean of 0 and sd of 1
colMeans(scaled.df2)  # faster version of apply(scaled.dat, 2, mean)
apply(scaled.df2, 2, sd)

k2 <- kmeans(df2,centers = 4)

par(mfrow= c(1,2))
plot(df1$X12,df1$X11, col=k1$cluster,frame = TRUE) # plot between Age and
Kg
points(k1$centers[,c(1,11)], col=1:4, pch=23, cex=4)

```

```
#autoplot(prcomp(df1),colour = k1$cluster) #PC plot

plot(df2$X1,df2$X11, col=k2$cluster,frame = TRUE) # plot between Age and Kg
points(k2$centers[,c(1,11)], col=1:4, pch=23, cex=4)

#autoplot(prcomp(df2),colour = k2$cluster) #PC plot
```

Appendix VII

R code for CPV Generation

```
library(ggplot2)
library(corrplot)
### CPV ###
NCustomers = 400
weights <- matrix(nrow = NCustomers,ncol = 15)
for (i in 1:NCustomers){
weights[i,] <- matrix(runif(15),ncol = 15)
}
weights = weights/rowSums(weights)

ratings <- matrix(nrow = NCustomers, ncol = 15)

for( i in 1:NCustomers){
  ratings[i,]  <- matrix(sample(1:10,15,replace = T))
}

CPV = rowSums(ratings*weights)
data1 <- data.frame(ratings,CPV)
Satistfaction = CPV>6
Satistfaction = as.integer(as.logical(Satistfaction))
CPV <- as.vector(CPV)
ratings <- data.frame(ratings)

y1 <- ratings[,1]
y2 <- ratings[,2]

colnames(data1) <-
c("1a","2a","3","4","5","6","7","8","9","10","11","12","13","14","15","Val"
)
df <- data.frame(CPV,y1,y2)

ggplot(df,aes(CPV, y = value, color = "variable"),geom=
c("point","smooth"),method = "lm", formula = y~x) +
   geom_smooth(aes(y= y1),color = "blue") +
   geom_smooth(aes(y= y2),color = "red") +
  geom_smooth(aes(y= ratings[,3]),color = "black") +
  geom_smooth(aes(y= ratings[,4]),color = "green") +
  geom_smooth(aes(y= ratings[,5]),color = "grey")

ratings <- as.data.frame(ratings)
corrplot(ratings, diag = FALSE, method="color", order="FPC", tl.srt = 90)

```

```
### plot x = city, Mountain, Village  ###
df <- read.csv("SimulatedData_400.csv")

## mean of city ratings ##
City_mean <- df[,"City"]*CPV
nCity <- sum(df[,"City"]>0)
City_mean <- sum(City_mean)/nCity


Mountain_mean <- df[,"Mountain"]*CPV
nMountain <- sum(df[,"Mountain"]>0)
Mountain_mean <- sum(Mountain_mean)/nMountain

Village_mean <- df[,"Village"]*CPV
nVillage <- sum(df[,"Village"]>0)
Village_mean <- sum(Village_mean)/nVillage

City_mean_factor1 <- df[,"City"]*data1[,"1a"]
nCity <- sum(df[,"City"]>0)
City_mean_factor1 <- sum(City_mean_factor1)/nCity


Mountain_mean_factor1 <- df[,"Mountain"]*data1[,"1a"]
nMountain <- sum(df[,"Mountain"]>0)
Mountain_mean_factor1 <- sum(Mountain_mean_factor1)/nMountain

Village_mean_factor1 <- df[,"Village"]*data1[,"1a"]
nVillage <- sum(df[,"Village"]>0)
Village_mean_factor1 <- sum(Village_mean_factor1)/nVillage

City_mean_factor2 <- df[,"City"]*data1[,"3"]
nCity <- sum(df[,"City"]>0)
City_mean_factor2 <- sum(City_mean_factor2)/nCity


Mountain_mean_factor2 <- df[,"Mountain"]*data1[,"3"]
nMountain <- sum(df[,"Mountain"]>0)
Mountain_mean_factor2 <- sum(Mountain_mean_factor2)/nMountain

Village_mean_factor2 <- df[,"Village"]*data1[,"3"]
nVillage <- sum(df[,"Village"]>0)
Village_mean_factor2 <- sum(Village_mean_factor2)/nVillage

# The plot
df1 <- data.frame(
```

```
  Rating_Type =
factor(c("Overall","Overall","Overall","Factor1","Factor1","Factor1","Facto
r2","Factor2","Factor2")),
  time =
factor(c("City","Village","Mountain","City","Village","Mountain","City","Vi
llage","Mountain"),
                levels=c("City","Village","Mountain")),
  Overall <- c(City_mean,Mountain_mean,Village_mean,
               City_mean_factor1,Mountain_mean_factor1,

Village_mean_factor1,City_mean_factor2,Mountain_mean_factor2,
               Village_mean_factor2)
)

# A basic graph
lp <- ggplot(data=df1, aes(x=time, y=Overall, group=Rating_Type,
 8                          shape=Rating_Type,color = Rating_Type))+
  geom_line() + geom_point() + labs(x="", y = "Ratings")
lp
```