CHAPTER I

INTRODUCTION

1.1 Scope of analysis

- A Chinese automobile company Geely-Auto's aspires to enter the US market by setting up their managing unit there and producing cars. As a statistical consultant working for an automobile industry, your task is to develop a model to predict the selling price of a car in US Market. Geely Company hopes to use this information to help assess the factors affecting the pricing of cars in the American marketing.

- Many factors, in addition, to predict the selling price of Cars. Among these, the influence of Company-name, model-name, and fuel-type, engine location, fuel system, highway or city mpg has been highlighted in the diverse field of cars research**.**

1.2 Approach of Analysis

With the increasing number of cars sold day by day, it has become difficult to manage (or) extract useful information from the available data of all the cars. The Geely_data is utilized to keep the data related to cars price prediction. This data is then used for visualizing the specifications of cars which is available in US market.

Additionally, the data is used to predict the selling price of the cars through various machine learning approaches. The proposed tool can prove beneficial for the Geely management in making of cars at US country.

CHAPTER II

DATA UNDERSTANDING

HR ANALYTICS

2.1 Gathering Data

Load the relevant Packages

```
```{r}
library(ggplot2)
library(tidyverse)
library(readr)
```
```

Load the dataset

```
```{r}
HR <- read_csv("D:/desktop/HR Analytics.csv")
View(HR)
```
```

Structure of the data

```
```{r}
str(HR)
```
```

```
tibble [9,011 x 18] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ SLNO                       : num [1:9011] 1 2 3 4 5 6 7 9 11 12 ...
 $ Candidate.Ref              : num [1:9011] 2110407 2112635 2112838 2115021 2115125 ...
 $ DOJ.Extended               : chr [1:9011] "Yes" "No" "No" "No" ...
 $ Duration.to.accept.offer   : num [1:9011] 14 18 3 26 1 17 37 16 1 6 ...
 $ Notice.period              : num [1:9011] 30 30 45 30 120 30 30 0 30 30 ...
 $ Offered.band               : chr [1:9011] "E2" "E2" "E2" "E2" ...
 $ Pecent.hike.expected.in.CTC: num [1:9011] -20.8 50 42.8 42.8 42.6 ...
 $ Percent.hike.offered.in.CTC: num [1:9011] 13.2 320 42.8 42.8 42.6 ...
 $ Percent.difference.CTC     : num [1:9011] 42.9 180 0 0 0 ...
 $ Joining.Bonus              : chr [1:9011] "No" "No" "No" "No" ...
 $ Candidate.relocate.actual  : chr [1:9011] "No" "No" "No" "No" ...
 $ Gender                     : chr [1:9011] "Female" "Male" "Male" "Male" ...
 $ Candidate.Source           : chr [1:9011] "Agency" "Employee Referral" "Agency" "Employee Referral" ...
 $ Rex.in.Yrs                 : num [1:9011] 7 8 4 4 6 2 7 8 3 3 ...
 $ LOB                        : chr [1:9011] "ERS" "INFRA" "INFRA" "INFRA" ...
 $ Location                   : chr [1:9011] "Noida" "Chennai" "Noida" "Noida" ...
 $ Age                        : num [1:9011] 34 34 27 34 34 34 32 34 26 34 ...
 $ Status                     : chr [1:9011] "Joined" "Joined" "Joined" "Joined" ...
```

2.2 Data description

The HR dataset has 9011 rows and 18 columns

SLNO	Candidate_Ref	DOJ_Extended	Duration_to_accept_offer	Notice_period	Offered_band	Pecent_hike_expected_in_CTC	Percen
1	2110407	Yes	14	30	E2	-20.79	
2	2112635	No	18	30	E2	50.00	
3	2112838	No	3	45	E2	42.84	
4	2115021	No	26	30	E2	42.84	
5	2115125	Yes	1	120	E2	42.59	
6	2117167	Yes	17	30	E1	42.83	
7	2119124	Yes	37	30	E2	31.58	
9	2127572	Yes	16	0	E1	-20.00	
11	2138169	No	1	30	E1	-22.22	
12	2143362	No	6	30	E1	240.00	
13	2151180	No	120	30	E2	5.26	
14	2154264	No	3	0	E2	28.21	
15	2156236	Yes	14	30	E2	50.00	
16	2158703	No	44	75	E2	45.45	
17	2161257	No	7	30	E3	53.85	
18	2162487	No	1	30	E3	-27.31	
19	2166041	Yes	98	30	E2	50.00	
20	2172982	No	1	0	E2	30.00	
21	2173730	No	1	30	E1	-13.42	
22	2175237	No	7	30	E1	221.43	
23	2175323	No	1	0	E3	42.78	
24	2184108	No	0	30	E1	37.50	
25	2188014	No	1	0	E2	42.86	
26	2191237	Yes	83	60	E2	25.00	
27	2194323	No	1	0	E3	114.82	
28	2205244	No	16	60	E1	41.18	
29	2205496	Yes	32	120	E3	50.97	
30	2207823	Yes	19	30	E2	42.85	
31	2211699	Yes	0	0	E2	23.08	
32	2214217	No	4	30	E2	-2.17	

The data-set contains the following Variables:

Candidate reference number

Unique number to identify the candidate

DOJ extended

Binary variable identifying whether candidate asked for date of joining extension (Yes/No)

Duration to accept the offer

Number of days taken by the candidate to accept the offer (continuous variable)

Notice period

Notice period to be served in the parting company before candidate can join this company (continuous variable)

Offered band

Band offered to the candidate based on experience and performance in interview rounds (categorical variable labelled C0/C1/C2/C3/C4/C5/C6)

Percentage hike (CTC) expected

Percentage hike expected by the candidate (continuous variable)

Percentage hike offered (CTC)

Percentage hike offered by the company (continuous variable)

Joining bonus

Binary variable indicating if joining bonus was given or not (Yes/No)

Gender

Gender of the candidate (Male/Female)

Candidate source

Source from which resume of the candidate was obtained (categorical variables with categories: Employee referral/Agency/Direct)

REX (in years)

Relevant years of experience of the candidate for the position offered (continuous variable)

LOB

Line of business for which offer was rolled out (categorical variable)

DOB

Date of birth of the candidate

Joining location

Company location for which offer was rolled out for candidate to join (categorical variable)

Candidate relocation status

Binary variable indicating whether candidate has to relocate from one city to another city for joining (Yes/No)

HR status

Final joining status of candidate (Joined/Not-Joined)

2.3 Data Understanding

This module explains data understanding. This dataset consist of different columns. Each and every columns we should find the summary () function. This function is used to calculate the average value and determine the maximum, minimum of the column in a data frame.

```
```{r}
summary(HR)
```

      SLNO   Candidate.Ref   DOJ.Extended   Duration.to.accept.offer Notice.period   Offered.band
 Min.   :   1   Min.   :2109586   Length:9011   Min.   :-228.00   Min.   : 0.00   Length:9011
 1st Qu.: 3202   1st Qu.:2383377   Class :character   1st Qu.:   3.00   1st Qu.: 30.00   Class :character
 Median : 5971   Median :2807385   Mode :character   Median : 10.00   Median : 30.00   Mode :character
 Mean   : 5968   Mean   :2843302   Mean   : 21.37   Mean   : 39.28
 3rd Qu.: 8736   3rd Qu.:3300058   3rd Qu.: 33.00   3rd Qu.: 60.00
 Max.   :12333   Max.   :3836076   Max.   : 224.00   Max.   :120.00
 Pecent.hike.expected.in.CTC Percent.hike.offered.in.CTC Percent.difference.CTC Joining.Bonus   Candidate.relocate.actual
 Min.   :-68.83   Min.   :-60.53   Min.   :-67.270   Length:9011   Length:9011
 1st Qu.: 27.27   1st Qu.: 22.08   1st Qu.: -8.330   Class :character   Class :character
 Median : 40.00   Median : 36.00   Median : 0.000   Mode :character   Mode :character
 Mean   : 43.86   Mean   : 40.66   Mean   : -1.569
 3rd Qu.: 53.85   3rd Qu.: 50.00   3rd Qu.: 0.000
 Max.   :359.77   Max.   :471.43   Max.   :300.000
    Gender   Candidate.Source   Rex.in.Yrs   LOB   Location   Age
 Length:9011   Length:9011   Min.   : 0.00   Length:9011   Length:9011   Min.   :20.00
 Class :character   Class :character   1st Qu.: 3.00   Class :character   Class :character   1st Qu.:27.00
 Mode :character   Mode :character   Median : 4.00   Mode :character   Mode :character   Median :29.00
                                       Mean   : 4.24   Mean   :29.92
                                       3rd Qu.: 6.00   3rd Qu.:34.00
                                       Max.   :24.00   Max.   :62.00
    Status
 Length:9011
 Class :character
 Mode :character
```

To show the categories of Categorical variable :

```{r}
table(DOJ.Extended)
table(Offered.band)
table(Joining.Bonus)
table(Candidate.relocate.actual)
table(Candidate.Source)
table(LOB)
table(Location)
table(Status)
```

```
DOJ.Extended
  No  Yes 
4802 4209 
Offered.band
  E0   E1   E2   E3 
 211 5578 2717  505 
Joining.Bonus
  No  Yes 
8593  418 
Candidate.relocate.actual
  No  Yes 
7717 1294 
Candidate.Source
           Agency            Direct Employee Referral 
             2587              4808              1616 
LOB
      AXON       BFSI       CSMP        EAS        ERS        ETS Healthcare      INFRA        MMS 
       569       1396        580        346       2427        691        126       2861         15 
Location
Ahmedabad Bangalore   Chennai    Cochin   Gurgaon Hyderabad   Kolkata    Mumbai     Noida    Others      Pune 
        6      2234      3152         8       147       341       129       198      2735        13        48 
Status
     Joined Not Joined 
       7326       1685 
```

NA’s in the dataset

```{r}
colSums(sapply(HR,is.na))
```

```
                     SLNO              Candidate.Ref              DOJ.Extended    Duration.to.accept.offer 
                        0                          0                         0                           0 
            Notice.period               Offered.band Pecent.hike.expected.in.CTC Percent.hike.offered.in.CTC 
                        0                          0                         0                           0 
    Percent.difference.CTC              Joining.Bonus  Candidate.relocate.actual                      Gender 
                        0                          0                         0                           0 
         Candidate.Source                 Rex.in.Yrs                        LOB                    Location 
                        0                          0                         0                           0 
                      Age                     Status 
                        0                          0 
```

Converting to factor

```{r}
HR$DOJ.Extended=as.factor(HR$DOJ.Extended)
HR$Offered.band=as.factor(HR$Offered.band)
HR$Candidate.relocate.actual=as.factor(HR$Candidate.relocate.actual)
HR$Candidate.Source=as.factor(HR$Candidate.Source)
HR$LOB=as.factor(HR$LOB)
HR$Gender=as.factor(HR$Gender)
HR$Location=as.factor(HR$Location)
HR$Status=as.factor(HR$Status)
```

UBER

2.1 Gathering Data

Load the relevant Packages

```
```{r}
library(ggplot2)
library(readr)
library(tidyverse)
```
```

Load the dataset

```
```{r}
library(readr)
uber=read_csv("C:/Users/makesh/Desktop/ranj mam/2. Uber Request Data.csv")
View(uber)
```
```

NA's in the dataset

```
```{r}
colSums(sapply(uber,is.na))
```
```

```
       Request id      Pickup point         Driver id            Status Request timestamp    Drop timestamp
                0                 0              2650                 0                 0                 0
```

Structure of the data

```
```{r}
str(uber)
```
```

```
tibble [6,745 x 6] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Request id       : num [1:6745] 619 867 1807 2532 3112 ...
 $ Pickup point     : chr [1:6745] "Airport" "Airport" "City" "Airport" ...
 $ Driver id        : num [1:6745] 1 1 1 1 1 1 1 1 1 2 ...
 $ Status           : chr [1:6745] "Trip Completed" "Trip Completed" "Trip Completed" "Trip Completed" ...
 $ Request timestamp: chr [1:6745] "11/7/2016 11:51" "11/7/2016 17:57" "12/7/2016 9:17" "12/7/2016 21:08" ...
 $ Drop timestamp   : chr [1:6745] "11/7/2016 13:00" "11/7/2016 18:47" "12/7/2016 9:58" "12/7/2016 22:03" ...
 - attr(*, "spec")=
  .. cols(
  ..   `Request id` = col_double(),
  ..   `Pickup point` = col_character(),
  ..   `Driver id` = col_double(),
  ..   Status = col_character(),
  ..   `Request timestamp` = col_character(),
  ..   `Drop timestamp` = col_character()
  .. )
```

2.2 Data description

Uber dataset has 6745 rows and 8columns

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
1	619	Airport	1	Trip Completed	11/7/2016 11:51	11/7/2016 13:00
2	867	Airport	1	Trip Completed	11/7/2016 17:57	11/7/2016 18:47
3	1807	City	1	Trip Completed	12/7/2016 9:17	12/7/2016 9:58
4	2532	Airport	1	Trip Completed	12/7/2016 21:08	12/7/2016 22:03
5	3112	City	1	Trip Completed	13-07-2016 08:33:16	13-07-2016 09:25:47
6	3879	Airport	1	Trip Completed	13-07-2016 21:57:28	13-07-2016 22:28:59
7	4270	Airport	1	Trip Completed	14-07-2016 06:15:32	14-07-2016 07:13:15
8	5510	Airport	1	Trip Completed	15-07-2016 05:11:52	15-07-2016 06:07:52
9	6248	City	1	Trip Completed	15-07-2016 17:57:27	15-07-2016 18:50:51
10	267	City	2	Trip Completed	11/7/2016 6:46	11/7/2016 7:25
11	1467	Airport	2	Trip Completed	12/7/2016 5:08	12/7/2016 6:02
12	1983	City	2	Trip Completed	12/7/2016 12:30	12/7/2016 12:57
13	2784	Airport	2	Trip Completed	13-07-2016 04:49:20	13-07-2016 05:23:03
14	3075	City	2	Trip Completed	13-07-2016 08:02:53	13-07-2016 09:16:19
15	3379	City	2	Trip Completed	13-07-2016 14:23:02	13-07-2016 15:35:18
16	3482	Airport	2	Trip Completed	13-07-2016 17:23:18	13-07-2016 18:20:51
17	4652	City	2	Trip Completed	14-07-2016 12:01:02	14-07-2016 12:36:46
18	5335	Airport	2	Trip Completed	14-07-2016 22:24:13	14-07-2016 23:18:52
19	535	Airport	3	Trip Completed	11/7/2016 10:00	11/7/2016 10:31
20	960	Airport	3	Trip Completed	11/7/2016 18:45	11/7/2016 19:23
21	1934	Airport	3	Trip Completed	12/7/2016 11:17	12/7/2016 12:23
22	2083	Airport	3	Trip Completed	12/7/2016 15:46	12/7/2016 16:40
23	2211	Airport	3	Trip Completed	12/7/2016 18:00	12/7/2016 18:28
24	3096	Airport	3	Trip Completed	13-07-2016 08:17:29	13-07-2016 09:22:37
25	3881	Airport	3	Trip Completed	13-07-2016 21:54:18	13-07-2016 22:51:23
26	5254	City	3	Trip Completed	14-07-2016 21:23:03	14-07-2016 22:25:19

The data-set contains the following Variables:

Request id:

A unique identifier of the request

Time of request:

The date and time at which the customer made the trip request

Drop-off time:

The drop-off date and time, in case the trip was completed

Pick-up point:

The point from which the request was made

Driver id:

The unique identification number of the driver 6. Status of the request: The final status of the trip that can be either completed, cancelled by the driver or no cars available

2.3 Data Understanding

This module explains data understanding. This dataset consist of different columns. Each and every columns we should find the summary () function. This function is used to calculate the average value and determine the maximum, minimum of the column in a data frame.

```{r}
summary(uber)
```

```
  Request id    Pickup point          Driver id         Status           Request timestamp   Drop timestamp
 Min.   :   1   Length:6745        Min.   :  1.0   Length:6745        Length:6745        Length:6745
 1st Qu.:1691   Class :character   1st Qu.: 75.0   Class :character   Class :character   Class :character
 Median :3387   Mode  :character   Median :149.0   Mode  :character   Mode  :character   Mode  :character
 Mean   :3385                      Mean   :149.5
 3rd Qu.:5080                      3rd Qu.:224.0
 Max.   :6766                      Max.   :300.0
                                   NA's   :2650
```

Data Cleaning

Separate the Date and time

```{r}
uber=uber%>%separate(Request_timestamp,into = c("request_date","request_time"),sep=" ")
uber=uber%>%separate(Drop_timestamp,into = c("drop_date","drop_time"),sep=" ")
```

Change the drive id value to Not_available

```{r}
uber = uber %>% mutate(drop_date = ifelse(is.na(drop_date),"Not_available",uber$drop_date))
uber = uber %>% mutate(drop_time = ifelse(is.na(drop_time),"Not_available",uber$drop_time))
```

```{r}
uber<-uber %>% mutate(Driver_id = ifelse(is.na(Driver_id), "Not_available",uber$Driver_id))
```

GRAMENER

2.1 Gathering Data

Load the relevant Packages

```
```{r}
library(tidyverse)
library(ggplot2)
library(readr)
```
```

Load the dataset

```
```{r}
gramener<- read_csv("C:/Users/makesh/Desktop/ranj mam/Gramener_Data.csv")
```
```

2.2 Data description

Gramener dataset has 39717 rows and 111 columns

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
1	1077501	1296599	5000	5000	4975.00	36 months	10.65%	162.87	B	B2	NA
2	1077430	1314167	2500	2500	2500.00	60 months	15.27%	59.83	C	C4	Ryder
3	1077175	1313524	2400	2400	2400.00	36 months	15.96%	84.33	C	C5	NA
4	1076863	1277178	10000	10000	10000.00	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD
5	1075358	1311748	3000	3000	3000.00	60 months	12.69%	67.79	B	B5	University Medical Group
6	1075269	1311441	5000	5000	5000.00	36 months	7.90%	156.46	A	A4	Veolia Transportaton
7	1069639	1304742	7000	7000	7000.00	60 months	15.96%	170.08	C	C5	Southern Star Photography
8	1072053	1288686	3000	3000	3000.00	36 months	18.64%	109.43	E	E1	MKC Accounting
9	1071795	1306957	5600	5600	5600.00	60 months	21.28%	152.39	F	F2	NA
10	1071570	1306721	5375	5375	5350.00	60 months	12.69%	121.45	B	B5	Starbucks
11	1070078	1305201	6500	6500	6500.00	60 months	14.65%	153.45	C	C3	Southwest Rural metro
12	1069908	1305008	12000	12000	12000.00	36 months	12.69%	402.54	B	B5	UCLA
13	1064687	1298717	9000	9000	9000.00	36 months	13.49%	305.38	C	C1	Va. Dept of Conservation/Recreation
14	1069866	1304956	3000	3000	3000.00	36 months	9.91%	96.68	B	B1	Target
15	1069057	1303503	10000	10000	10000.00	36 months	10.65%	325.74	B	B2	SFMTA
16	1069759	1304871	1000	1000	1000.00	36 months	16.29%	35.31	D	D1	Internal revenue Service
17	1065775	1299699	10000	10000	10000.00	36 months	15.27%	347.98	C	C4	Chin's Restaurant
18	1069971	1304884	3600	3600	3600.00	36 months	6.03%	109.57	A	A1	Duracell
19	1062474	1294539	6000	6000	6000.00	36 months	11.71%	198.46	B	B3	Connection Inspection
20	1069742	1304855	9200	9200	9200.00	36 months	6.03%	280.01	A	A1	Network Interpreting Service
21	1069740	1284848	20250	20250	19142.16	60 months	15.27%	484.63	C	C4	Archdiocese of Galveston Houston
22	1039153	1269083	21000	21000	21000.00	36 months	12.42%	701.73	B	B4	Osram Sylvania
23	1069710	1304821	10000	10000	10000.00	36 months	11.71%	330.76	B	B3	Value Air
24	1069700	1304810	10000	10000	10000.00	36 months	11.71%	330.76	B	B3	Wells Fargo Bank
25	1069559	1304634	6000	6000	6000.00	36 months	11.71%	198.46	B	B3	bmg-educational
26	1069697	1273773	15000	15000	15000.00	36 months	9.91%	483.38	B	B1	Winfield Pathology Consultants

Customer's Demographic Information

- Emp_title
- Emp_length
- Home_ownership
- Annual_inc
- Verification_status
- Addr_state
- Zip_code
- Title
- Purpose
- Desc
- url

Loan Characteristics Information

- Loan amount
- Funded amount
- Funded amount invested
- Interest rate
- Loan status
- Loan grade
- Loan sub_grade
- Dti
- Loan issue date
- Loan term
- Installment

Credit information: Customer Behavior Variable

- Delinq_2yrs
- Earliest_cr_line
- Inq_last_6mths
- Open_acc
- Pub_rec
- Revol_bal
- Revol_util
- Total_acc
- Out_prncp
- Out_prncp_inv
- Total_pymnt
- Total_pymnt_inv
- Total_rec_prncp
- Total_rec_int
- Total_rec_late_fee
- Recoveries
- Collection_recovery_fee

- Last_pymnt_d
- Last_pymnt_amnt
- Next_pymnt_d
- Last_credit_pull_d
- Application_type

NA’s in the dataset

```{r}
colSums(sapply(gramener,is.na))
```

```
                     id                member_id                 loan_amnt              funded_amnt
                      0                        0                         0                        0
        funded_amnt_inv                     term                  int_rate              installment
                      0                        0                         0                        0
                  grade                sub_grade                 emp_title               emp_length
                      0                        0                      2453                        0
         home_ownership               annual_inc       verification_status                  issue_d
                      0                        0                         0                        0
            loan_status               pymnt_plan                       url                     desc
                      0                        0                         0                    13149
                purpose                    title                  zip_code               addr_state
                      0                       10                         0                        0
                    dti              delinq_2yrs          earliest_cr_line           inq_last_6mths
                      0                        0                         0                        0
 mths_since_last_delinq   mths_since_last_record                  open_acc                  pub_rec
                  25682                    36931                         0                        0
              revol_bal               revol_util                 total_acc      initial_list_status
                      0                       50                         0                        0
              out_prncp            out_prncp_inv               total_pymnt          total_pymnt_inv
                      0                        0                         0                        0
        total_rec_prncp            total_rec_int        total_rec_late_fee               recoveries
                      0                        0                         0                        0
collection_recovery_fee             last_pymnt_d           last_pymnt_amnt             next_pymnt_d
                      0                       71                         0                    38577
     last_credit_pull_d collections_12_mths_ex_med mths_since_last_major_derog          policy_code
                      2                       56                     39717                        0
       application_type         annual_inc_joint                 dti_joint verification_status_joint
                      0                    39717                     39717                    39717
         acc_now_delinq             tot_coll_amt               tot_cur_bal              open_acc_6m
                      0                    39717                     39717                    39717
             open_il_6m              open_il_12m               open_il_24m       mths_since_rcnt_il
                  39717                    39717                     39717                    39717
           total_bal_il                  il_util               open_rv_12m              open_rv_24m
                  39717                    39717                     39717                    39717
             max_bal_bc                 all_util          total_rev_hi_lim                   inq_fi
                  39717                    39717                     39717                    39717
            total_cu_tl             inq_last_12m      acc_open_past_24mths              avg_cur_bal
                  39717                    39717                     39717                    39717
```

2.3 Data Understanding

This module explains data understanding. This dataset consist of different columns. Each and every columns we should find the summary () function. This function is used to calculate the average value and determine the maximum, minimum of the column in a data frame.

```{r}
summary(gramener)
```

```
       id           member_id         loan_amnt      funded_amnt    funded_amnt_inv      term             int_rate        
 Min.   :  54734   Min.   :  70699   Min.   :  500   Min.   :  500   Min.   :    0   Length:39717       Length:39717      
 1st Qu.: 516221   1st Qu.: 666780   1st Qu.: 5500   1st Qu.: 5400   1st Qu.: 5000   Class :character   Class :character  
 Median : 665665   Median : 850812   Median :10000   Median : 9600   Median : 8975   Mode  :character   Mode  :character  
 Mean   : 683132   Mean   : 850464   Mean   :11219   Mean   :10948   Mean   :10397                                        
 3rd Qu.: 837755   3rd Qu.:1047339   3rd Qu.:15000   3rd Qu.:15000   3rd Qu.:14400                                        
 Max.   :1077501   Max.   :1314167   Max.   :35000   Max.   :35000   Max.   :35000                                        

  installment         grade            sub_grade          emp_title          emp_length        home_ownership    
 Min.   :  15.69   Length:39717       Length:39717       Length:39717       Length:39717       Length:39717      
 1st Qu.: 167.02   Class :character   Class :character   Class :character   Class :character   Class :character  
 Median : 280.22   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character  
 Mean   : 324.56                                                                                                 
 3rd Qu.: 430.78                                                                                                 
 Max.   :1305.19                                                                                                 

   annual_inc      verification_status   issue_d          loan_status         pymnt_plan            url           
 Min.   :   4000   Length:39717        Length:39717       Length:39717       Length:39717       Length:39717      
 1st Qu.:  40404   Class :character    Class :character   Class :character   Class :character   Class :character  
 Median :  59000   Mode  :character    Mode  :character   Mode  :character   Mode  :character   Mode  :character  
 Mean   :  68969                                                                                                  
 3rd Qu.:  82300                                                                                                  
 Max.   :6000000                                                                                                  

     desc              purpose             title             zip_code          addr_state             dti       
 Length:39717       Length:39717       Length:39717       Length:39717       Length:39717       Min.   : 0.00  
 Class :character   Class :character   Class :character   Class :character   Class :character   1st Qu.: 8.17  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Median :13.40  
                                                                                                Mean   :13.32  
                                                                                                3rd Qu.:18.60  
                                                                                                Max.   :29.99  
```

Variable names:

```{r}
names(gramener)
```

```
  [1] "id"                             "member_id"                      "loan_amnt"                     
  [4] "funded_amnt"                    "funded_amnt_inv"                "term"                          
  [7] "int_rate"                       "installment"                    "grade"                         
 [10] "sub_grade"                      "emp_title"                      "emp_length"                    
 [13] "home_ownership"                 "annual_inc"                     "verification_status"           
 [16] "issue_d"                        "loan_status"                    "pymnt_plan"                    
 [19] "url"                            "desc"                           "purpose"                       
 [22] "title"                          "zip_code"                       "addr_state"                    
 [25] "dti"                            "delinq_2yrs"                    "earliest_cr_line"              
 [28] "inq_last_6mths"                 "mths_since_last_delinq"         "mths_since_last_record"        
 [31] "open_acc"                       "pub_rec"                        "revol_bal"                     
 [34] "revol_util"                     "total_acc"                      "initial_list_status"           
 [37] "out_prncp"                      "out_prncp_inv"                  "total_pymnt"                   
 [40] "total_pymnt_inv"                "total_rec_prncp"                "total_rec_int"                 
 [43] "total_rec_late_fee"             "recoveries"                     "collection_recovery_fee"       
 [46] "last_pymnt_d"                   "last_pymnt_amnt"                "next_pymnt_d"                  
 [49] "last_credit_pull_d"             "collections_12_mths_ex_med"     "mths_since_last_major_derog"   
 [52] "policy_code"                    "application_type"               "annual_inc_joint"              
 [55] "dti_joint"                      "verification_status_joint"      "acc_now_delinq"                
 [58] "tot_coll_amt"                   "tot_cur_bal"                    "open_acc_6m"                   
 [61] "open_il_6m"                     "open_il_12m"                    "open_il_24m"                   
 [64] "mths_since_rcnt_il"             "total_bal_il"                   "il_util"                       
 [67] "open_rv_12m"                    "open_rv_24m"                    "max_bal_bc"                    
 [70] "all_util"                       "total_rev_hi_lim"               "inq_fi"                        
 [73] "total_cu_tl"                    "inq_last_12m"                   "acc_open_past_24mths"          
 [76] "avg_cur_bal"                    "bc_open_to_buy"                 "bc_util"                       
 [79] "chargeoff_within_12_mths"       "delinq_amnt"                    "mo_sin_old_il_acct"            
 [82] "mo_sin_old_rev_tl_op"           "mo_sin_rcnt_rev_tl_op"          "mo_sin_rcnt_tl"                
 [85] "mort_acc"                       "mths_since_recent_bc"           "mths_since_recent_bc_dlq"      
 [88] "mths_since_recent_inq"          "mths_since_recent_revol_delinq" "num_accts_ever_120_pd"         
 [91] "num_actv_bc_tl"                 "num_actv_rev_tl"                "num_bc_sats"                   
 [94] "num_bc_tl"                      "num_il_tl"                      "num_op_rev_tl"                 
 [97] "num_rev_accts"                  "num_rev_tl_bal_gt_0"            "num_sats"                      
[100] "num_tl_120dpd_2m"               "num_tl_30dpd"                   "num_tl_90g_dpd_24m"            
[103] "num_tl_op_past_12m"             "pct_tl_nvr_dlq"                 "percent_bc_gt_75"              
[106] "pub_rec_bankruptcies"           "tax_liens"                      "tot_hi_cred_lim"               
[109] "total_bal_ex_mort"              "total_bc_limit"                 "total_il_high_credit_limit"    
```

unselect the unwanted columns

````
```{r}
gramener=select(gramener,-total_il_high_credit_limit,-total_bc_limit,-total_bal_ex_mort,-tot_hi_cred_lim,-percent_bc_gt_75,-pct_tl
_nvr_dlq,-num_tl_op_past_12m,num_tl_90g_dpd_24m,num_tl_30dpd,num_tl_120dpd_2m,num_sats,num_rev_tl_bal_gt_0,num_rev_accts,num_op_re
v_tl)
```
```{r}
gramener=gramener%>%select(-(tot_coll_amt:bc_util))
```
```{r}
gramener=gramener%>%select(-(mo_sin_old_il_acct:num_tl_90g_dpd_24m))
```
```{r}
gramener=gramener%>%select(-mths_since_last_major_derog,-annual_inc_joint,-verification_status_joint,-dti_joint)
```
````

````
```{r}
gramener<-gramener%>%
  mutate(desc= if_else(is.na(desc), "nodesc", as.character(desc)),
        mths_since_last_delinq=if_else(is.na(mths_since_last_delinq), "no_mths_since_last_charge",
as.character(mths_since_last_delinq)),
        mths_since_last_record=if_else(is.na(mths_since_last_record), "no", as.character(mths_since_last_record)),
        last_pymnt_d=if_else(is.na(last_pymnt_d), "nolastpayment", as.character(last_pymnt_d)),
        next_pymnt_d=if_else(is.na(next_pymnt_d), "nopayment", as.character(next_pymnt_d)),
        last_credit_pull_d=if_else(is.na(last_credit_pull_d), "no", as.character(last_credit_pull_d)),
        collections_12_mths_ex_med=if_else(is.na(collections_12_mths_ex_med), "no", as.character(collections_12_mths_ex_med)),
        chargeoff_within_12_mths=if_else(is.na(chargeoff_within_12_mths), "no", as.character(chargeoff_within_12_mths)),
        pub_rec_bankruptcies=if_else(is.na(pub_rec_bankruptcies), "no", as.character(pub_rec_bankruptcies)),
        tax_liens=if_else(is.na(tax_liens), "no", as.character(tax_liens)),
        title=if_else(is.na(title), "no-title", as.character(title)),
        emp_title=if_else(is.na(emp_title), "no_emp_title", as.character(emp_title)),
        revol_util=if_else(is.na(revol_util), "no_revol", as.character(revol_util)))
```
````

Check the NA’S

````
```{r}
colSums(sapply(gramener,is.na))
```
````

```
                      id                member_id                loan_amnt              funded_amnt
                       0                        0                        0                        0
         funded_amnt_inv                     term                 int_rate              installment
                       0                        0                        0                        0
                   grade                sub_grade                emp_title               emp_length
                       0                        0                        0                        0
          home_ownership               annual_inc      verification_status                  issue_d
                       0                        0                        0                        0
             loan_status               pymnt_plan                      url                     desc
                       0                        0                        0                        0
                 purpose                    title                 zip_code               addr_state
                       0                        0                        0                        0
                     dti              delinq_2yrs         earliest_cr_line           inq_last_6mths
                       0                        0                        0                        0
  mths_since_last_delinq   mths_since_last_record                 open_acc                  pub_rec
                       0                        0                        0                        0
               revol_bal               revol_util                total_acc      initial_list_status
                       0                        0                        0                        0
               out_prncp            out_prncp_inv              total_pymnt          total_pymnt_inv
                       0                        0                        0                        0
         total_rec_prncp            total_rec_int       total_rec_late_fee               recoveries
                       0                        0                        0                        0
 collection_recovery_fee             last_pymnt_d          last_pymnt_amnt             next_pymnt_d
                       0                        0                        0                        0
      last_credit_pull_d collections_12_mths_ex_med              policy_code         application_type
                       0                        0                        0                        0
          acc_now_delinq   chargeoff_within_12_mths              delinq_amnt     pub_rec_bankruptcies
                       0                        0                        0                        0
               tax_liens
                       0
```

Structure of the dataset

````
```{r}
str(gramener)
```
````

```
tibble [39,717 x 57] (S3: tbl_df/tbl/data.frame)
 $ id                         : num [1:39717] 1077501 1077430 1077175 1076863 1075358 ...
 $ member_id                  : num [1:39717] 1296599 1314167 1313524 1277178 1311748 ...
 $ loan_amnt                  : num [1:39717] 5000 2500 2400 10000 3000 ...
 $ funded_amnt                : num [1:39717] 5000 2500 2400 10000 3000 ...
 $ funded_amnt_inv            : num [1:39717] 4975 2500 2400 10000 3000 ...
 $ term                       : chr [1:39717] "36 months" "60 months" "36 months" "36 months" ...
 $ int_rate                   : chr [1:39717] "10.65%" "15.27%" "15.96%" "13.49%" ...
 $ installment                : num [1:39717] 162.9 59.8 84.3 339.3 67.8 ...
 $ grade                      : chr [1:39717] "B" "C" "C" "C" ...
 $ sub_grade                  : chr [1:39717] "B2" "C4" "C5" "C1" ...
 $ emp_title                  : chr [1:39717] "no_emp_title" "Ryder" "no_emp_title" "AIR RESOURCES BOARD" ...
 $ emp_length                 : chr [1:39717] "10+ years" "< 1 year" "10+ years" "10+ years" ...
 $ home_ownership             : chr [1:39717] "RENT" "RENT" "RENT" "RENT" ...
 $ annual_inc                 : num [1:39717] 24000 30000 12252 49200 80000 ...
 $ verification_status        : chr [1:39717] "Verified" "Source Verified" "Not Verified" "Source Verified" ...
 $ issue_d                    : chr [1:39717] "Dec-11" "Dec-11" "Dec-11" "Dec-11" ...
 $ loan_status                : chr [1:39717] "Fully Paid" "Charged Off" "Fully Paid" "Fully Paid" ...
 $ pymnt_plan                 : chr [1:39717] "n" "n" "n" "n" ...
 $ url                        : chr [1:39717] "https://lendingclub.com/browse/loanDetail.action?loan_id=1077501"
"https://lendingclub.com/browse/loanDetail.action?loan_id=1077430" "https://lendingclub.com/browse/loanDetail.action?
loan_id=1077175" "https://lendingclub.com/browse/loanDetail.action?loan_id=1076863" ...
 $ desc                       : chr [1:39717] "Borrower added on 12/22/11 > I need to upgrade my business technologies.<br>"
"Borrower added on 12/22/11 > I plan to use this money to finance the motorcycle i am looking at. I plan to have"| __truncated__
"nodesc" "Borrower added on 12/21/11 > to pay for property tax (borrow from friend, need to pay back) & central A/C need "|
__truncated__ ...
 $ purpose                    : chr [1:39717] "credit_card" "car" "small_business" "other" ...
 $ title                      : chr [1:39717] "Computer" "bike" "real estate business" "personel" ...
 $ zip_code                   : chr [1:39717] "860xx" "309xx" "606xx" "917xx" ...
 $ addr_state                 : chr [1:39717] "AZ" "GA" "IL" "CA" ...
 $ dti                        : num [1:39717] 27.65 1 8.72 20 17.94 ...
 $ delinq_2yrs                : num [1:39717] 0 0 0 0 0 0 0 0 0 0 ...
 $ earliest_cr_line           : chr [1:39717] "Jan-85" "Apr-99" "Nov-01" "Feb-96" ...
 $ inq_last_6mths             : num [1:39717] 1 5 2 1 0 3 1 2 2 0 ...
 $ mths_since_last_delinq     : chr [1:39717] "no_mths_since_last_charge" "no_mths_since_last_charge" "no_mths_since_last_charge"
"35" ...
```

To check the categories and count of the categories of the variable

````
## Exploratory Data Analysis

```{r}
table(addr_state)
```
````

```
addr_state
  AK   AL   AR   AZ   CA   CO   CT   DC   DE   FL   GA   HI   IA   ID   IL   IN   KS   KY   LA   MA   MD   ME   MI   MN   MO 
  80  452  245  879 7099  792  751  214  114 2866 1398  174    5    6 1525    9  271  325  436 1340 1049    3  720  615  686 
  MS   MT   NC   NE   NH   NJ   NM   NV   NY   OH   OK   OR   PA   RI   SC   SD   TN   TX   UT   VA   VT   WA   WI   WV   WY 
  19   85  788    5  171 1850  189  497 3812 1223  299  451 1517  198  472   64   17 2727  258 1407   54  840  460  177   83 
```

````
```{r}
table(grade)
```
````

```
grade
    A     B     C     D     E     F     G 
10085 12020  8098  5307  2842  1049   316 
```

````
```{r}
table(term)
```
````

```
term
36 months 60 months 
    29096     10621 
```

```{r}
table(sub_grade)
table(loan_status)
```

```
sub_grade
  A1   A2   A3   A4   A5   B1   B2   B3   B4   B5   C1   C2   C3   C4   C5   D1   D2   D3   D4   D5   E1   E2   E3   E4   E5 
1139 1508 1810 2886 2742 1830 2057 2917 2512 2704 2136 2011 1529 1236 1186  931 1348 1173  981  874  763  656  553  454  416 
  F1   F2   F3   F4   F5   G1   G2   G3   G4   G5 
 329  249  185  168  118  104   78   48   56   30 
loan_status
Charged Off     Current  Fully Paid 
       5627        1140       32950 
```

```{r}
gramener$loan_status=as.factor(gramener$loan_status)
gramener$sub_grade=as.factor(gramener$sub_grade)
gramener$term=as.factor(gramener$term)
gramener$grade=as.factor(gramener$grade)
gramener$addr_state=as.factor(addr_state)
```

GEELY

2.1 Gathering Data

Load the relevant Packages

```{r}
library(ggplot2)
library(MASS)
library(tree)
library(randomForest)
library(Hmisc)
library(tidyverse)
library(DescTools)
```

Load the dataset

```{r}
library(readr)
geely<-  read.csv("C:/Users/makesh/Downloads/CarPrice_Assignment.csv")
View(geely)
```

Structure of the data

```{r}
str(geely)
```

```
'data.frame':	205 obs. of  26 variables:
 $ car_ID          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ symboling       : int  3 3 1 2 2 2 1 1 1 0 ...
 $ CarName         : chr  "alfa-romero giulia" "alfa-romero stelvio" "alfa-romero Quadrifoglio" "audi 100 ls" ...
 $ fueltype        : chr  "gas" "gas" "gas" "gas" ...
 $ aspiration      : chr  "std" "std" "std" "std" ...
 $ doornumber      : chr  "two" "two" "two" "four" ...
 $ carbody         : chr  "convertible" "convertible" "hatchback" "sedan" ...
 $ drivewheel      : chr  "rwd" "rwd" "rwd" "fwd" ...
 $ enginelocation  : chr  "front" "front" "front" "front" ...
 $ wheelbase       : num  88.6 88.6 94.5 99.8 99.4 ...
 $ carlength       : num  169 169 171 177 177 ...
 $ carwidth        : num  64.1 64.1 65.5 66.2 66.4 66.3 71.4 71.4 71.4 67.9 ...
 $ carheight       : num  48.8 48.8 52.4 54.3 54.3 53.1 55.7 55.7 55.9 52 ...
 $ curbweight      : int  2548 2548 2823 2337 2824 2507 2844 2954 3086 3053 ...
 $ enginetype      : chr  "dohc" "dohc" "ohcv" "ohc" ...
 $ cylindernumber  : chr  "four" "four" "six" "four" ...
 $ enginesize      : int  130 130 152 109 136 136 136 136 131 131 ...
 $ fuelsystem      : chr  "mpfi" "mpfi" "mpfi" "mpfi" ...
 $ boreratio       : num  3.47 3.47 2.68 3.19 3.19 3.19 3.19 3.19 3.13 3.13 ...
 $ stroke          : num  2.68 2.68 3.47 3.4 3.4 3.4 3.4 3.4 3.4 3.4 ...
 $ compressionratio: num  9 9 9 10 8 8.5 8.5 8.5 8.3 7 ...
 $ horsepower      : int  111 111 154 102 115 110 110 110 140 160 ...
 $ peakrpm         : int  5000 5000 5000 5500 5500 5500 5500 5500 5500 5500 ...
 $ citympg         : int  21 21 19 24 18 19 19 19 17 16 ...
 $ highwaympg      : int  27 27 26 30 22 25 25 25 20 22 ...
 $ price           : num  13495 16500 16500 13950 17450 ...
```

2.2 Data description

The geely dataset has 205 rows 26 column

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase
1	1	3	alfa-romero giulia	gas	std	two	convertible	rwd	front	
2	2	3	alfa-romero stelvio	gas	std	two	convertible	rwd	front	
3	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	front	
4	4	2	audi 100 ls	gas	std	four	sedan	fwd	front	
5	5	2	audi 100ls	gas	std	four	sedan	4wd	front	
6	6	2	audi fox	gas	std	two	sedan	fwd	front	
7	7	1	audi 100ls	gas	std	four	sedan	fwd	front	
8	8	1	audi 5000	gas	std	four	wagon	fwd	front	
9	9	1	audi 4000	gas	turbo	four	sedan	fwd	front	
10	10	0	audi 5000s (diesel)	gas	turbo	two	hatchback	4wd	front	
11	11	2	bmw 320i	gas	std	two	sedan	rwd	front	
12	12	0	bmw 320i	gas	std	four	sedan	rwd	front	
13	13	0	bmw x1	gas	std	two	sedan	rwd	front	
14	14	0	bmw x3	gas	std	four	sedan	rwd	front	
15	15	1	bmw z4	gas	std	four	sedan	rwd	front	
16	16	0	bmw x4	gas	std	four	sedan	rwd	front	
17	17	0	bmw x5	gas	std	two	sedan	rwd	front	
18	18	0	bmw x3	gas	std	four	sedan	rwd	front	
19	19	2	chevrolet impala	gas	std	two	hatchback	fwd	front	
20	20	1	chevrolet monte carlo	gas	std	two	hatchback	fwd	front	
21	21	0	chevrolet vega 2300	gas	std	four	sedan	fwd	front	
22	22	1	dodge rampage	gas	std	two	hatchback	fwd	front	
23	23	1	dodge challenger se	gas	std	two	hatchback	fwd	front	
24	24	1	dodge d200	gas	turbo	two	hatchback	fwd	front	
25	25	1	dodge monaco (sw)	gas	std	four	hatchback	fwd	front	
26	26	1	dodge colt hardtop	gas	std	four	sedan	fwd	front	
27	27	1	dodge colt (sw)	gas	std	four	sedan	fwd	front	
28	28	1	dodge coronet custom	gas	turbo	two	sedan	fwd	front	
29	29	-1	dodge dart custom	gas	std	four	wagon	fwd	front	
30	30	3	dodge coronet custom (sw)	gas	turbo	two	hatchback	fwd	front	
31	31	2	honda civic	gas	std	two	hatchback	fwd	front	

The data-set contains the following Variables:

Car_ID

Unique id of each observation (Integer)

Symboling

Its assigned insurance risk rating, a value of +3 indicates that the auto is risky, -3 that it is probably pretty safe. (Categorical)

CarCompany

Name of car company (Categorical)

Fueltype

Car fuel type i.e. gas or diesel (Categorical)

Aspiration

Aspiration used in a car (Categorical)

Doornumber

Number of doors in a car (Categorical)

Carbody

Body of car (Categorical)

Drivewheel

Type of drive wheel (Categorical)

Enginelocation

Location of car engine (Categorical)

Wheelbase

Wheelbase of car (Numeric)

Carlength

Length of car (Numeric)

Carwidth

Width of car (Numeric)

Carheight

Height of car (Numeric)

Curbweight

The weight of a car without occupants or baggage. (Numeric)

Enginetype

Type of engine. (Categorical)

Cylindernumber

Cylinder placed in the car (Categorical)

Enginesize

Size of car (Numeric)

Fuelsystem

Fuel system of car (Categorical)

Boreratio

Boreratio of car (Numeric)

Stroke

Stroke or volume inside the engine (Numeric)

Compressionratio

Compression ratio of car (Numeric)

Horsepower

Horsepower (Numeric)

Peakrpm

Car peak rpm (Numeric)

Citympg

Mileage in city (Numeric)

Highwaympg

Mileage on highway (Numeric)

Price (Dependent variable)

Price of car (Numeric)

2.3 Data Understanding

This module explains data understanding. This dataset consist of different columns. Each and every columns we should find the summary () function. This function is used to calculate the average value and determine the maximum, minimum of the column in a data frame.

```
```{r}
summary(geely)
```
```

```
    car_ID        symboling          CarName            fueltype           aspiration         doornumber
 Min.   :  1   Min.   :-2.0000   Length:205         Length:205         Length:205         Length:205
 1st Qu.: 52   1st Qu.: 0.0000   Class :character   Class :character   Class :character   Class :character
 Median :103   Median : 1.0000   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Mean   :103   Mean   : 0.8341
 3rd Qu.:154   3rd Qu.: 2.0000
 Max.   :205   Max.   : 3.0000
   carbody            drivewheel         enginelocation       wheelbase        carlength        carwidth
 Length:205         Length:205         Length:205         Min.   : 86.60   Min.   :141.1   Min.   :60.30
 Class :character   Class :character   Class :character   1st Qu.: 94.50   1st Qu.:166.3   1st Qu.:64.10
 Mode  :character   Mode  :character   Mode  :character   Median : 97.00   Median :173.2   Median :65.50
                                                          Mean   : 98.76   Mean   :174.0   Mean   :65.91
                                                          3rd Qu.:102.40   3rd Qu.:183.1   3rd Qu.:66.90
                                                          Max.   :120.90   Max.   :208.1   Max.   :72.30
   carheight       curbweight    enginetype        cylindernumber       enginesize     fuelsystem           boreratio
 Min.   :47.80   Min.   :1488   Length:205         Length:205         Min.   : 61.0   Length:205         Min.   :2.54
 1st Qu.:52.00   1st Qu.:2145   Class :character   Class :character   1st Qu.: 97.0   Class :character   1st Qu.:3.15
 Median :54.10   Median :2414   Mode  :character   Mode  :character   Median :120.0   Mode  :character   Median :3.31
 Mean   :53.72   Mean   :2556                                         Mean   :126.9                      Mean   :3.33
 3rd Qu.:55.50   3rd Qu.:2935                                         3rd Qu.:141.0                      3rd Qu.:3.58
 Max.   :59.80   Max.   :4066                                         Max.   :326.0                      Max.   :3.94
     stroke      compressionratio   horsepower        peakrpm        citympg        highwaympg        price
 Min.   :2.070   Min.   : 7.00    Min.   : 48.0   Min.   :4150   Min.   :13.00   Min.   :16.00   Min.   : 5118
 1st Qu.:3.110   1st Qu.: 8.60    1st Qu.: 70.0   1st Qu.:4800   1st Qu.:19.00   1st Qu.:25.00   1st Qu.: 7788
 Median :3.290   Median : 9.00    Median : 95.0   Median :5200   Median :24.00   Median :30.00   Median :10295
 Mean   :3.255   Mean   :10.14    Mean   :104.1   Mean   :5125   Mean   :25.22   Mean   :30.75   Mean   :13277
 3rd Qu.:3.410   3rd Qu.: 9.40    3rd Qu.:116.0   3rd Qu.:5500   3rd Qu.:30.00   3rd Qu.:34.00   3rd Qu.:16503
 Max.   :4.170   Max.   :23.00    Max.   :288.0   Max.   :6600   Max.   :49.00   Max.   :54.00   Max.   :45400
```

Describe of dataset

```
```{r}
describe(geely)
```
```

```
geely

 26  Variables      205  Observations
---------------------------------------------------------------------------------------------------------------------------
car_ID
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
     205        0      205        1      103    68.67     11.2     21.4     52.0    103.0    154.0    184.6    194.8

lowest :   1   2   3   4   5, highest: 201 202 203 204 205
---------------------------------------------------------------------------------------------------------------------------
symboling
       n  missing distinct     Info     Mean      Gmd
     205        0        6     0.94   0.8341    1.383

lowest : -2 -1  0  1  2, highest: -1  0  1  2  3

Value         -2    -1     0     1     2     3
Frequency      3    22    67    54    32    27
Proportion 0.015 0.107 0.327 0.263 0.156 0.132
---------------------------------------------------------------------------------------------------------------------------
```

NA's in the dataset

```r
colSums(sapply(geely,is.na))
sum(is.na(geely))
```

```
          car_ID        symboling          CarName         fueltype       aspiration       doornumber          carbody 
               0                0                0                0                0                0                0 
       drivewheel   enginelocation        wheelbase        carlength         carwidth        carheight       curbweight 
               0                0                0                0                0                0                0 
       enginetype   cylindernumber       enginesize       fuelsystem        boreratio           stroke compressionratio 
               0                0                0                0                0                0                0 
       horsepower          peakrpm          citympg       highwaympg            price 
               0                0                0                0                0 
[1] 0
```

Fueltype:

Fueltype is a Categorical variable .There are two fuel type

- Diesel
- Gas

```r
table(fueltype)
```

```
fueltype
diesel    gas 
    20    185 
```

Aspiration :

Aspiration is a Categorical variable . It is divided into two

- Std
- Turbo

```r
table(aspiration)
```

```
aspiration
  std turbo 
  168    37 
```

Doornumber:

Doornumber is a Categorical variable . It is divided into two

- Four
- Two

```r
table(doornumber)
```

```
doornumber
four  two 
 115   90 
```

Carbody :

Carbody is a Categorical variable . It is divided into five

- Convertible
- Hardtop
- Hatchback
- Sedan
- Wagon

```{r}
table(carbody)
```

```
carbody
convertible     hardtop   hatchback       sedan       wagon
          6           8          70          96          25
```

Drivewheel:

Drivewheel is a Categorical variable . It is divided into three

- 4wd
- Fwd
- rwd

```{r}
table(drivewheel)
```

```
drivewheel
4wd fwd rwd
  9 120  76
```

Enginelocation :

Enginelocation is a Categorical variable . It is divided into two

- Front
- Rear

```{r}
table(enginelocation)
```

```
enginelocation
front  rear
  202     3
```

Enginetype :

Enginetype is a Categorical variable . It is divided into seven

- Dohc
- Dohcv
- L
- Ohc
- Ohcf
- Ohcv
- rotor

```{r}
table(enginetype)
```

```
enginetype
 dohc dohcv     l   ohc  ohcf  ohcv rotor
   12     1    12   148    15    13     4
```

Cylindernumber:

Cylinder number is a Categorical variable . It is divided into seven

- Eight
- Five
- Four
- Six
- Three
- Twelve
- Two

```{r}
table(cylindernumber)
```

```
cylindernumber
 eight   five   four    six  three twelve    two
     5     11    159     24      1      1      4
```

2.4 Data Cleaning

Changing the Cylinder number Categorical to numerical

```{r}
geely$cylindernumber[which(geely$cylindernumber=="four")] = 4
geely$cylindernumber[which(geely$cylindernumber=="six")] = 6
geely$cylindernumber[which(geely$cylindernumber=="five")] = 5
geely$cylindernumber[which(geely$cylindernumber=="eight")] = 8
geely$cylindernumber[which(geely$cylindernumber=="two")] = 2
geely$cylindernumber[which(geely$cylindernumber=="three")] = 3
geely$cylindernumber[which(geely$cylindernumber=="twelve")] = 12

```

Separating the Companyname and model

```{r}
library(tidyverse)
library(tidyr)
geely=geely%>%separate(CarName,into=c("company_name","model"),sep=" ")
```

Changing the companyname as factor

```{r}
geely$company_name=as.factor(geely$company_name)
```

Changing the company name

```{r}
table(geely$company_name)
```

```
 alfa-romero        audi         bmw       buick   chevrolet       dodge       honda       isuzu 
           3           7           8           8           3           9          13           4 
      jaguar       maxda       mazda     mercury  mitsubishi      nissan      Nissan     peugeot 
           3           0          17           1          13          18           0          11 
    plymouth    porcshce     porsche     renault        saab      subaru      toyota     toyouta 
           7           0           5           2           6          12          32           0 
   vokswagen  volkswagen       volvo          vw 
           0          12          11           0 
```

Convert the character Variable to factor

// convert the character variabls to factor

```{r}
geely$fueltype       = as.factor(geely$fueltype)
geely$aspiration     = as.factor(geely$aspiration)
geely$doornumber     = as.factor(geely$doornumber)
geely$carbody        = as.factor(geely$carbody)
geely$drivewheel     = as.factor(geely$drivewheel)
geely$enginelocation = as.factor(geely$enginelocation)
geely$enginetype     = as.factor(geely$enginetype)
geely$cylindernumber = as.factor(geely$cylindernumber)
geely$fuelsystem     =  as.factor(geely$fuelsystem)
```

Convert the door number categorical to numeric

```{r}
table(geely$doornumber) ### Need to change numbers
geely$doornumber = as.character(geely$doornumber)
geely$doornumber[which(geely$doornumber=="four")] = 4
geely$doornumber[which(geely$doornumber=="two")] = 2
geely$doornumber = as.integer(geely$doornumber)

```

```
  2   4 
 90 115 
```

CHAPTER III

EDA USING R

3.1 Exploratory Data Analysis

- When you first get your data, it’s very tempting to immediately begin fitting models and assessing how they perform. However, before you begin modeling, it’s absolutely essential to explore the structure of the data and the relationships between the variables in the data set.

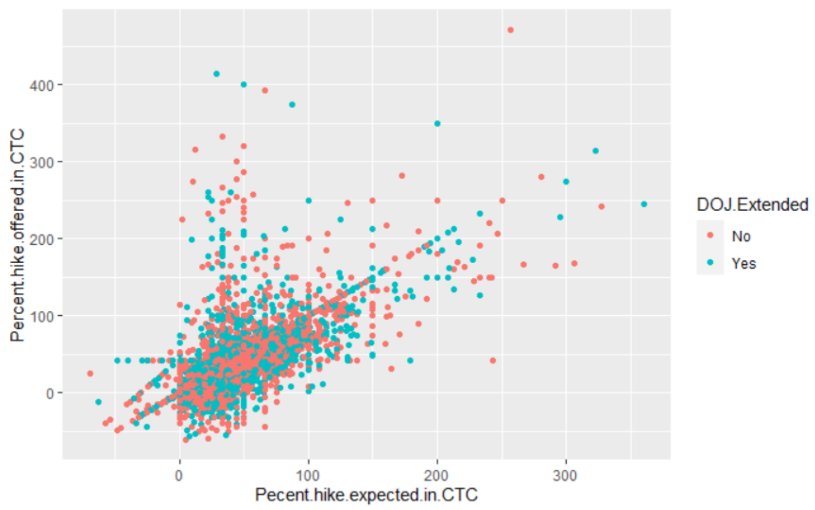
- Do a detailed EDA of the geely data set, to learn about the structure of the data and the relationships between the variables in the data set (refer to Data description sheet of geely data). Your EDA should involve creating and reviewing many plots/graphs and considering the patterns and relationships you see.

HR ANALYTICS

Plot 1

This plot clearly explains the companies offered jobs vs company expected jobs according to the DOJ of the job workers.

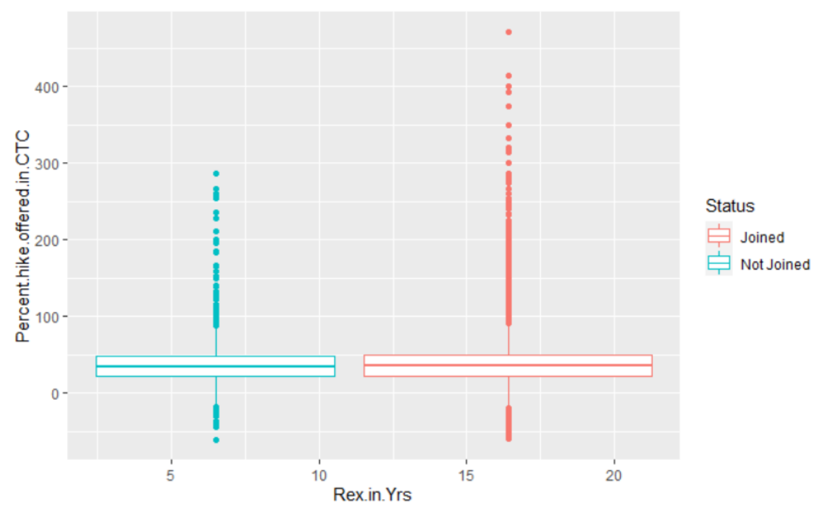
```
```{r}
ggplot(data=HR,mapping =aes(x=Pecent.hike.expected.in.CTC,y=Percent.hike.offered.in.CTC,col=DOJ.Extended))+geom_point()
```
```



Plot 2

Experience of candidates vs percent of the offered jobs .

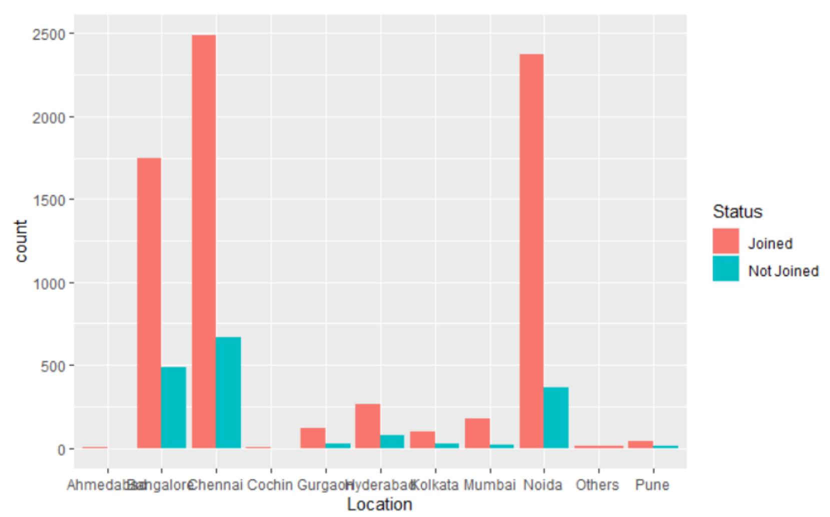
```{r}
ggplot(data=HR,mapping = aes(x=Rex.in.Yrs,y=Percent.hike.offered.in.CTC,col=Status))+geom_boxplot()
```



Plot 3

The count of joined not joined in each city.

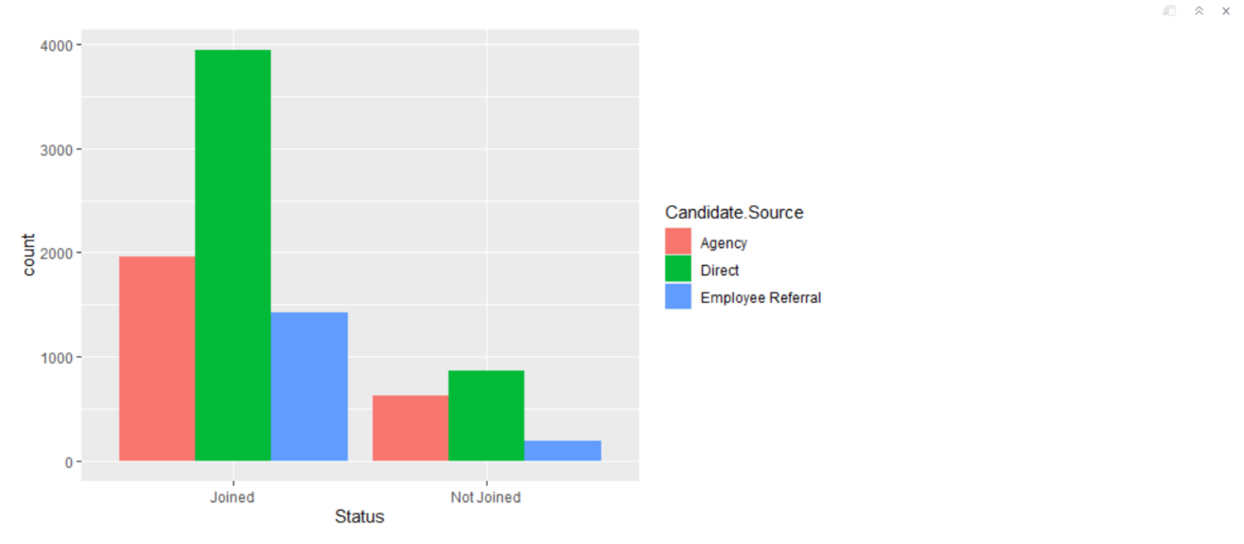
```{r}
ggplot(data=HR)+geom_bar(mapping = aes(Location,fill=Status),position="dodge")
```



Plot 4

The count of joined and joined workers who are all came from agency or direct or employee referral.

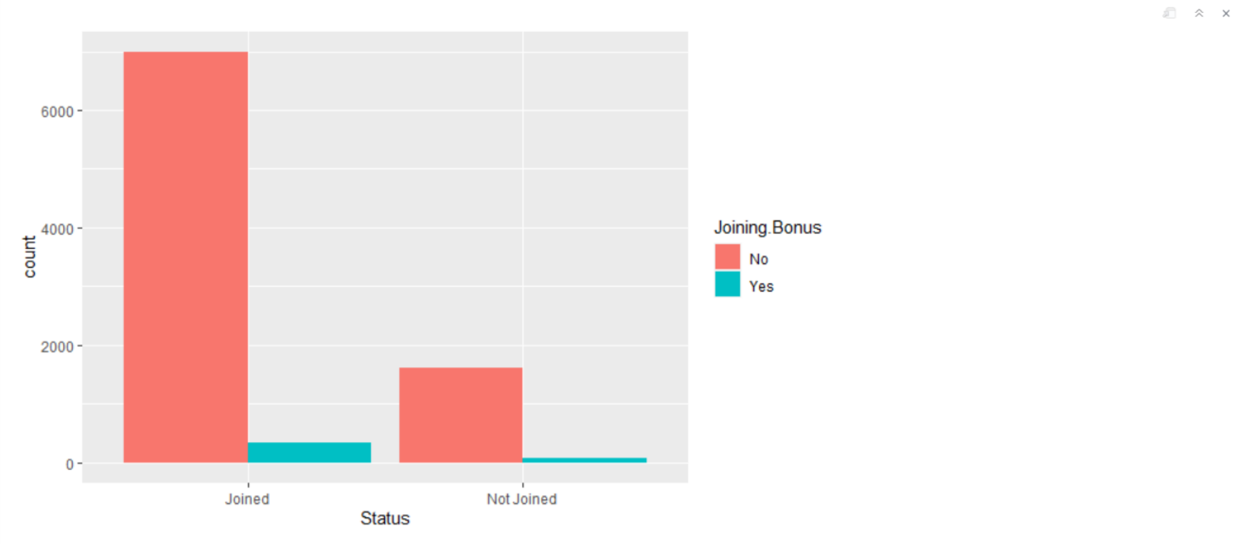
```{r}
ggplot(data=HR)+geom_bar(mapping = aes(Status,fill=Candidate.Source),position="dodge")
```



Plot 5

The count of joined and not joined according to the joining bonus.

```{r}
ggplot(data=HR)+geom_bar(mapping = aes(Status,fill=Joining.Bonus),position="dodge")
```

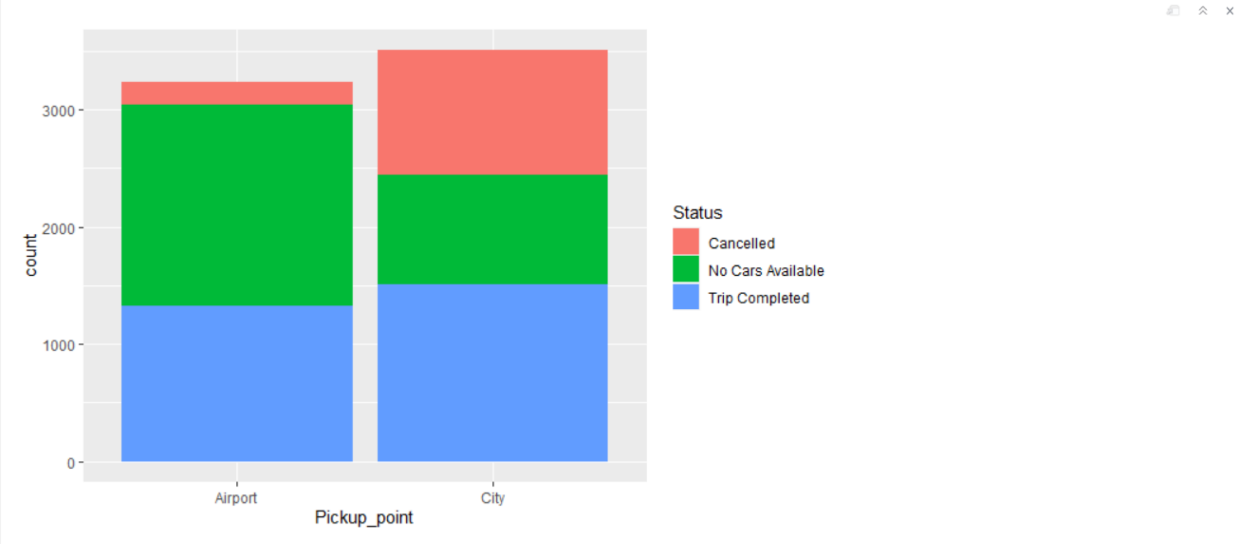


UBER

Plot 1

This plot clearly explains that the Pickup_Point count according to the count of Status which divided into three types [Cancelled , Trip Completed , No Cars Available] .

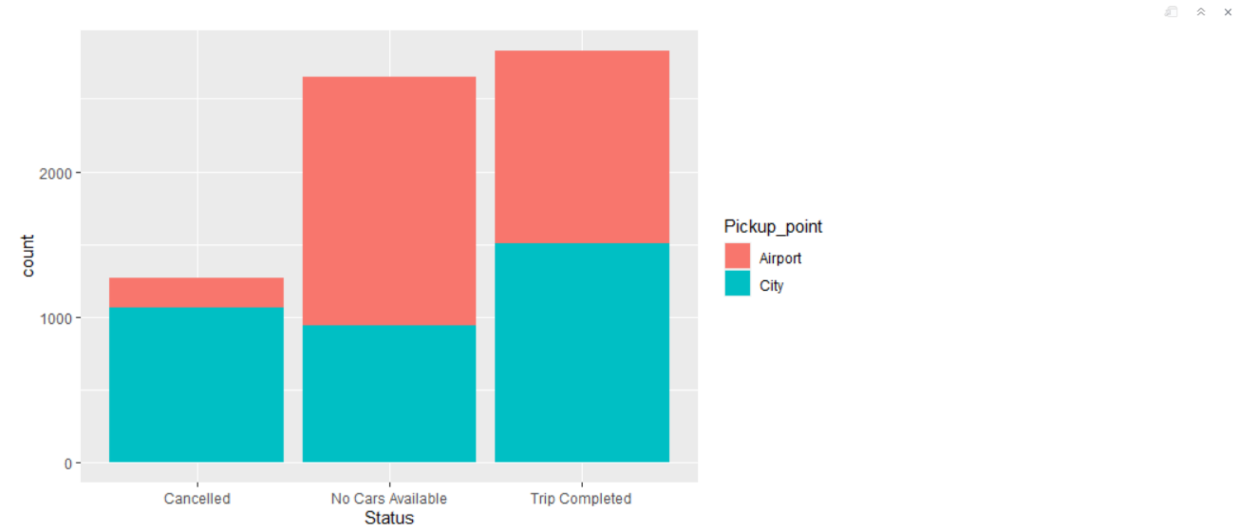
~~~
## EDA:
```{r}
uber%>%ggplot(mapping=aes(Pickup_point,fill=Status))+geom_bar()
```
~~~



Plot 2

This plot clearly explains that the Status count according to the count of Pickup_count which divided into two types [Airport , City] .

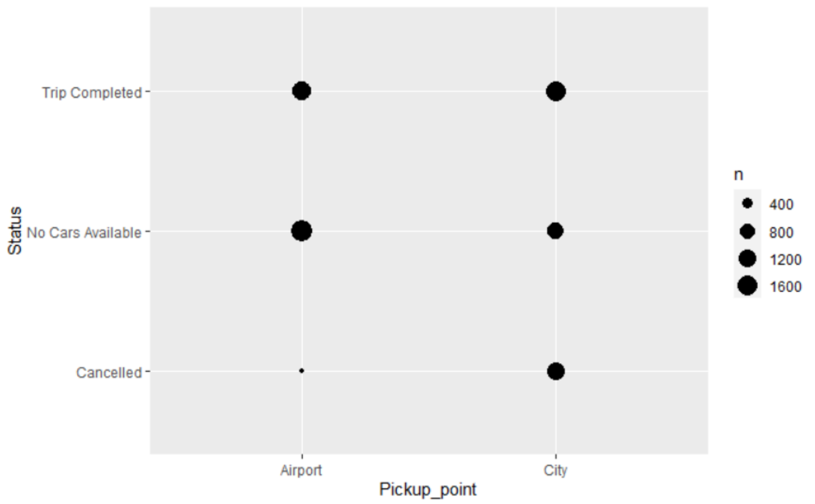
~~~
```{r}
uber%>%ggplot(mapping=aes(Status,fill=Pickup_point))+geom_bar()
```
~~~



Plot 3:

How many Cancelled and number of cars available in airport and city using   geom_count () plot.

```{r}
uber%>%ggplot(mapping=aes(Pickup_point,Status))+geom_count()
```

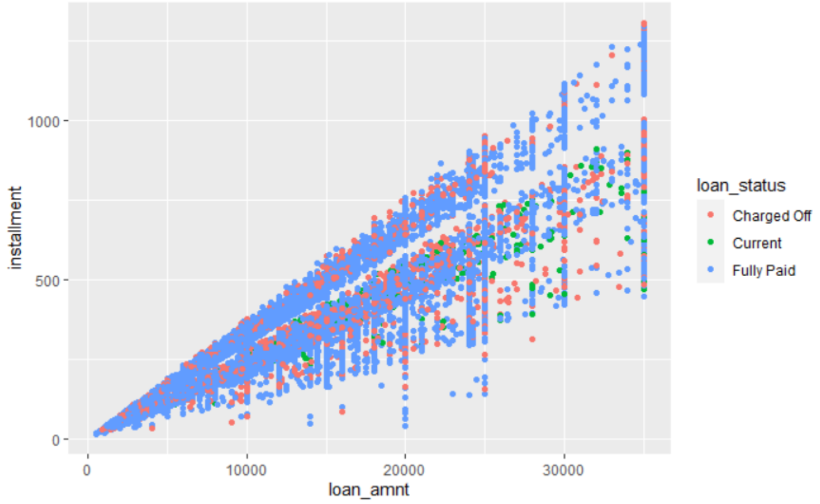


GRAMENER

Plot 1 :

Loan amount vs installment segregated according to loan_status

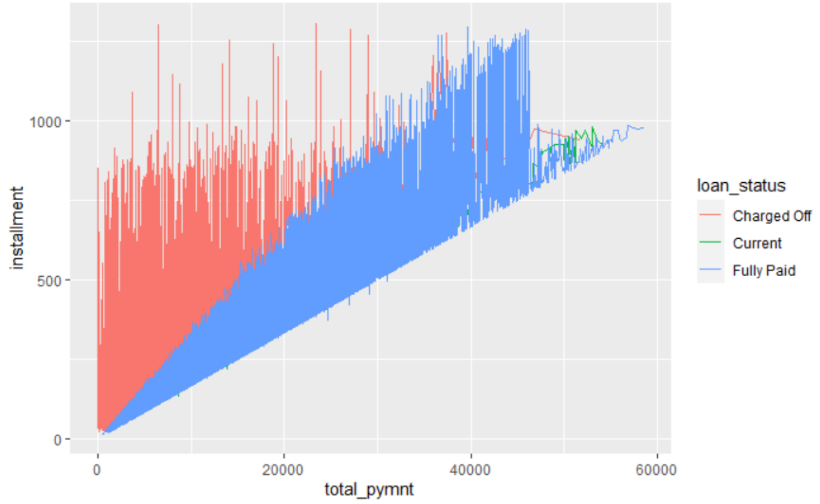
```{r}
ggplot(gramener,mapping=aes(loan_amnt,installment,col=loan_status))+geom_point()
```



Plot 2:

Total payment vs installment due according to loan_status .

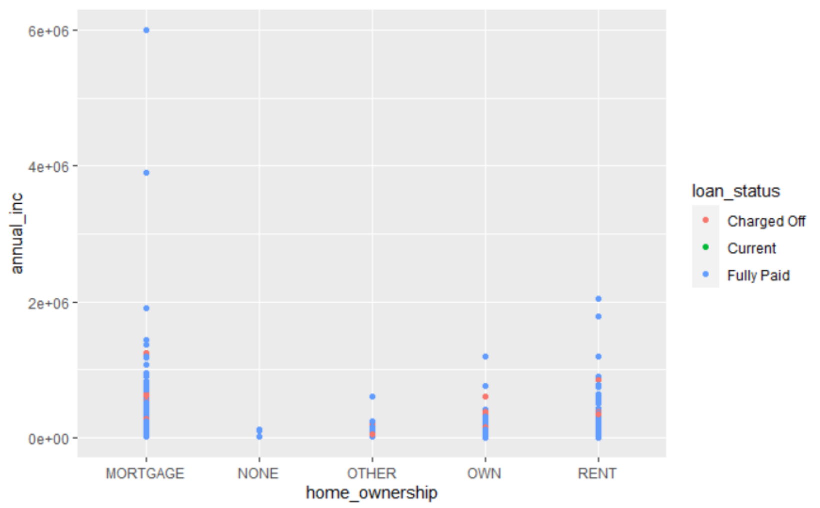
```{r}
ggplot(gramener,mapping=aes(total_pymnt,installment,col=loan_status))+geom_line()
```



Plot 3:

Annual income of home owners .

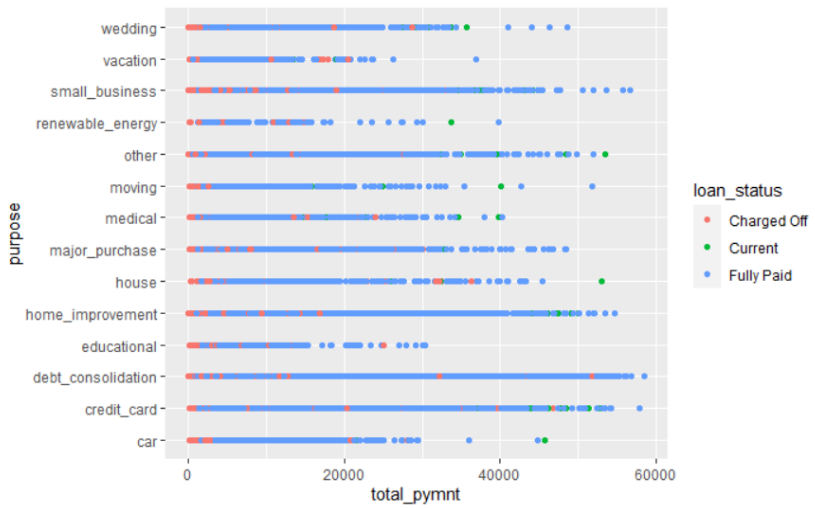
```{r}
ggplot(gramener,mapping=aes(home_ownership,annual_inc,col=loan_status))+geom_point()
```



Plot 4 :

To check whether the relationship between Total payment and purpose

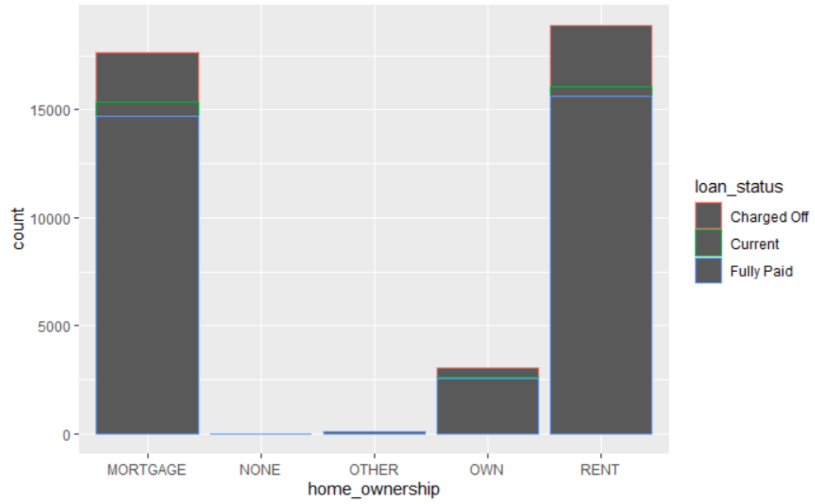
```{r}
ggplot(gramener,mapping=aes(total_pymnt,purpose,col=loan_status))+geom_point()
```



Plot 5 :

The count of home_ownership according to Mortgage ,None,other,own and rent .

```{r}
ggplot(gramener,mapping=aes(home_ownership,col=loan_status))+geom_bar()
```

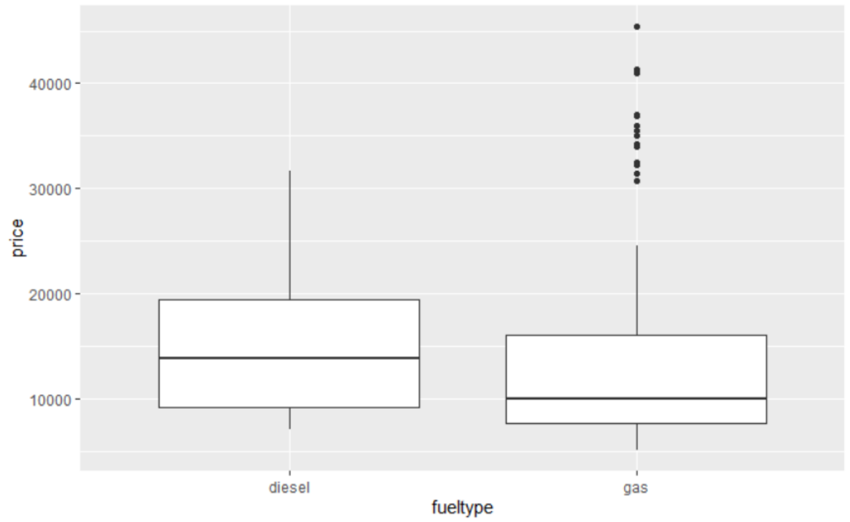


GEELY

Plot 1

To find fuel type & price have a relationship or not

~~~~
// geom_boxplot
```{r}
ggplot(geely,mapping=aes(fueltype,price))+geom_boxplot()
```
~~~~

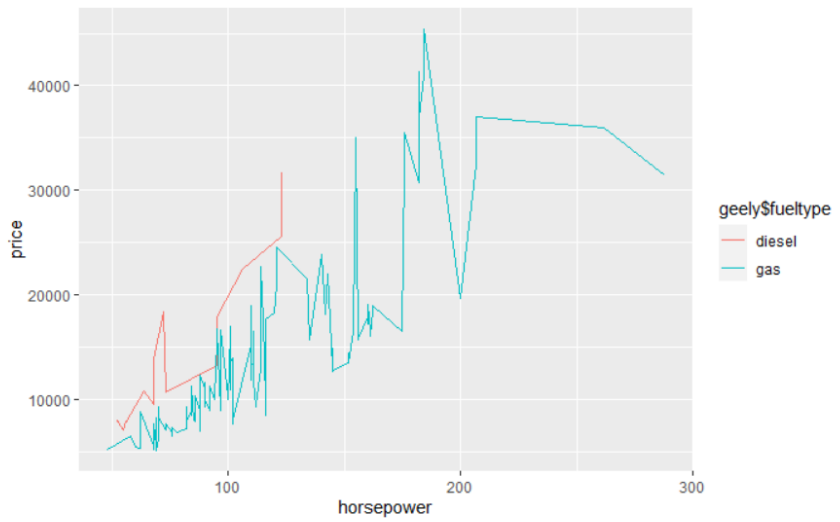


Plot 2

Horsepower (vs) price according to fueltype

~~~~
// Line Chart
```{r}
ggplot(geely,mapping=aes(horsepower,price,col=geely$fueltype))+geom_line()
```
~~~~

Use of `geely$fueltype` is discouraged. Use `fueltype` instead.

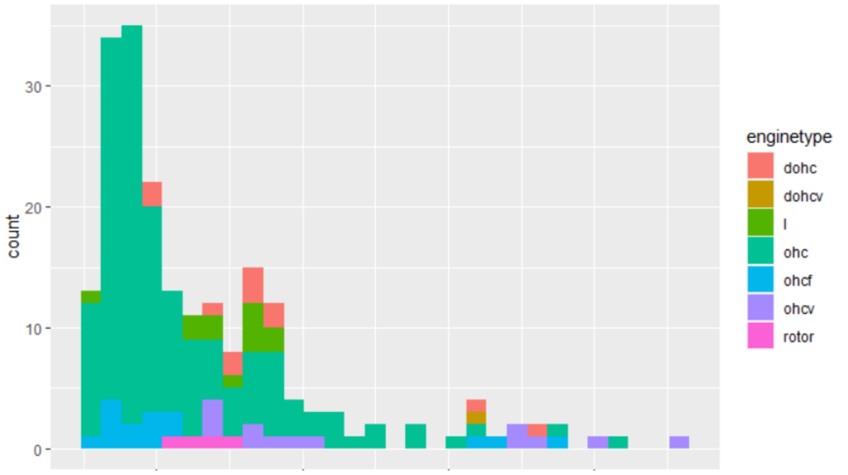


Plot 3

Engine type prices

```{r}
ggplot(geely)+geom_histogram(mapping=aes(fill=enginetype,price),Position = "dodge")
```

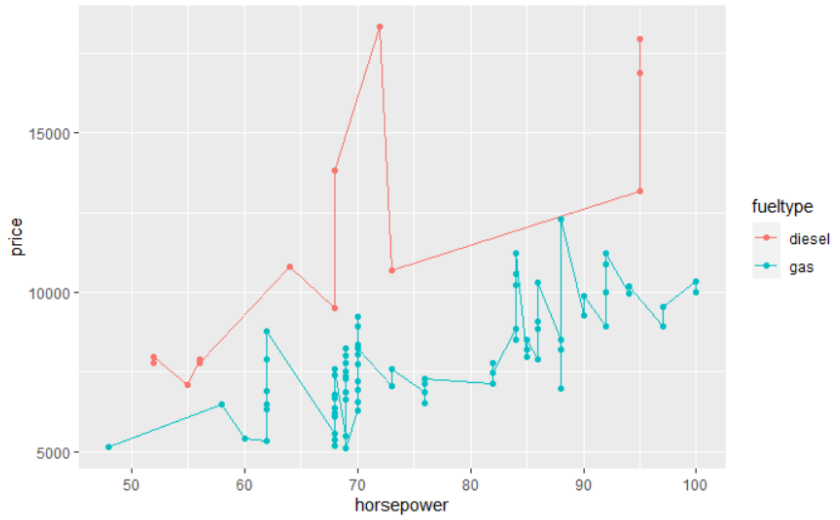
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



Plot 4

Horsepower (vs) price according to that fuel type

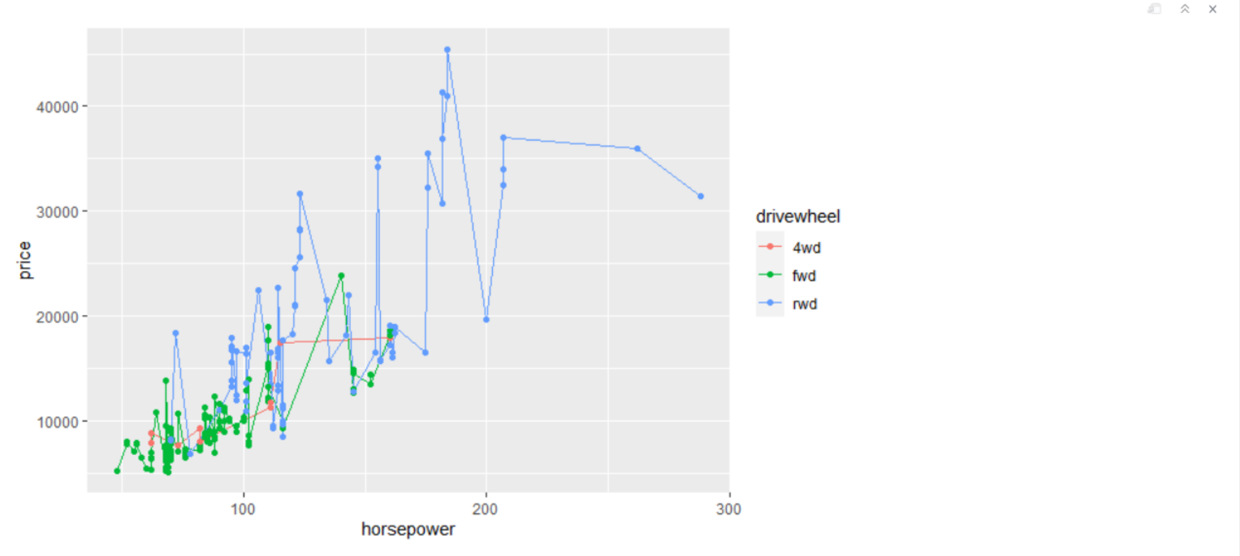
```{r}
geely%>%filter(highwaympg>30)%>%ggplot(mapping=aes(horsepower,price,col=fueltype))+geom_line()+geom_point()
```



Plot 5

Horsepower (vs) price according to that wheel type

```{r}
geely%>%filter()%>%ggplot(mapping=aes(horsepower,price,col=drivewheel))+geom_line()+geom_point()
```



CHAPTER IV

MODEL BUILDING

4.1 Choosing the model

Choosing a model to use is very essential. You must consider the input and output of your data. For this data:

- The data is labelled, so it's a supervised learning problem
- The data is looking to predict a number as output, so it's a regression problem
- So, now we will be looking for Regression model that works on supervised learning problems

Divide the dataset into Training data and Testing data

- Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing .
- After a model has been processed by using the training set, you test the model by making predictions against the test set.

```
```{r}
r=sample(nrow(geely),nrow(geely)*0.9)
train=geely[r,]
test=geely[-r,]
```
```

Linear Regression

- Linear regression may be defined as the statistical model that analyzes the linear relationship between a dependent variable with given set of independent variables.
- Does a set of predictor variables do a good job in predicting an outcome (dependent) variable.
- Which variables in particular are significant predictors of the outcome variable
- Three major uses for regression analysis are determining the strength of predictors, forecasting an effect, and trend forecasting.

```{r}
yup=lm(price~company_name,data =train)
summary(yup)
```

```
Call:
lm(formula = price ~ company_name, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-9910.7 -2081.5  -447.8  1344.7 14974.3 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)            15498.33    2134.00   7.263 1.51e-11 ***
company_nameaudi        2360.83    2550.62   0.926  0.35603    
company_namebmw        10842.38    2550.62   4.251 3.59e-05 ***
company_namebuick      17841.67    2613.61   6.826 1.66e-10 ***
company_namechevrolet  -9491.33    3017.94  -3.145  0.00198 ** 
company_namedodge      -7334.96    2502.34  -2.931  0.00386 ** 
company_namehonda      -7202.83    2385.89  -3.019  0.00295 ** 
company_nameisuzu      -6581.83    2823.02  -2.331  0.02096 *  
company_namejaguar     20051.67    4268.00   4.698 5.57e-06 ***
company_namemazda      -5257.73    2337.68  -2.249  0.02585 *  
company_namemercury     1004.67    4268.00   0.235  0.81420    
company_namemitsubishi -5937.67    2385.89  -2.489  0.01383 *  
company_namenissan     -5311.83    2325.48  -2.284  0.02366 *  
company_namepeugeot      -18.33    2433.14  -0.008  0.99400    
company_nameplymouth   -7534.90    2550.62  -2.954  0.00360 ** 
company_nameporsche    18247.79    2823.02   6.464 1.15e-09 ***
company_namerenault    -5903.33    3374.15  -1.750  0.08209 .  
company_namesaab        1331.67    2823.02   0.472  0.63776    
company_namesubaru     -6957.08    2385.89  -2.916  0.00405 ** 
company_nametoyota     -5372.87    2245.42  -2.393  0.01786 *  
company_namevolkswagen -5411.97    2407.48  -2.248  0.02593 *  
company_namevolvo       2564.85    2407.48   1.065  0.28830    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3696 on 162 degrees of freedom
Multiple R-squared:  0.7909,	Adjusted R-squared:  0.7638
```

From the model we have learnt BMW,buick,Chevrolet,dodge,iszu,jaguar,mitsubushi,Plymouth,Porsche,Subaru,toyato,Volkswagen are the significant values of the model

- If bmw value change it will affect in price 10842.38 increase
- If buick value change it will affect in price 17841.67 increase
- If chevrolet value change it will affect in price -9491.33 decrease
- If dodge value it will affect in price 1-7334.96 decrease
- If honda value it will affect in price -7202.83 decrease
- If jaguar value it will affect in price 20051.67 increase
- If porsche value it will affect in price 18247.79 increase
- If subaru value it will affect in price -6957.08 decrease
- If toyato value it will affect in price -5372.87 decrease
- If volkswagen value it will affect in price -5411.97 decrease

```{r}
str(model)
```

```
chr [1:205] "giulia" "stelvio" "Quadrifoglio" "100" "100ls" "fox" "100ls" "5000" "4000" "5000s" "320i" "320i" "x1" ...
```

```{r}
geely$model=as.factor(geely$model)
```

```{r}
tr=predict(yup,test)
tr
```

```
       15        22        32        48        50        57        59        70        74        77        95       105 
26340.714  8163.375  8295.500 35550.000 35550.000 10240.600 10240.600 33340.000 33340.000  9560.667 10186.500 10186.500 
      112       126       133       134       152       164       168       176       191 
15480.000 33746.125 16830.000 16830.000 10125.464 10125.464 10125.464 10125.464 10086.364 
```

```{r}
mean((tr-test$price)^2)
```

```
[1] 19999973
```

According to Linear regression error rate will be **19999973 .**

Decision Tree:

- Decision Tree Algorithm is a supervised Machine Learning Algorithm. It is an approach to predictive analysis that can help you make decisions. Decision tree goes down in a tree-structured format.
- Tree-based methods are simple and useful for interpretation. Decision trees can be applied to both regression and classification problems.
- There are three methods used in decision tree Bagging, random forests, and boosting. These methods grow multiple trees which are then combined to yield a single consensus prediction.

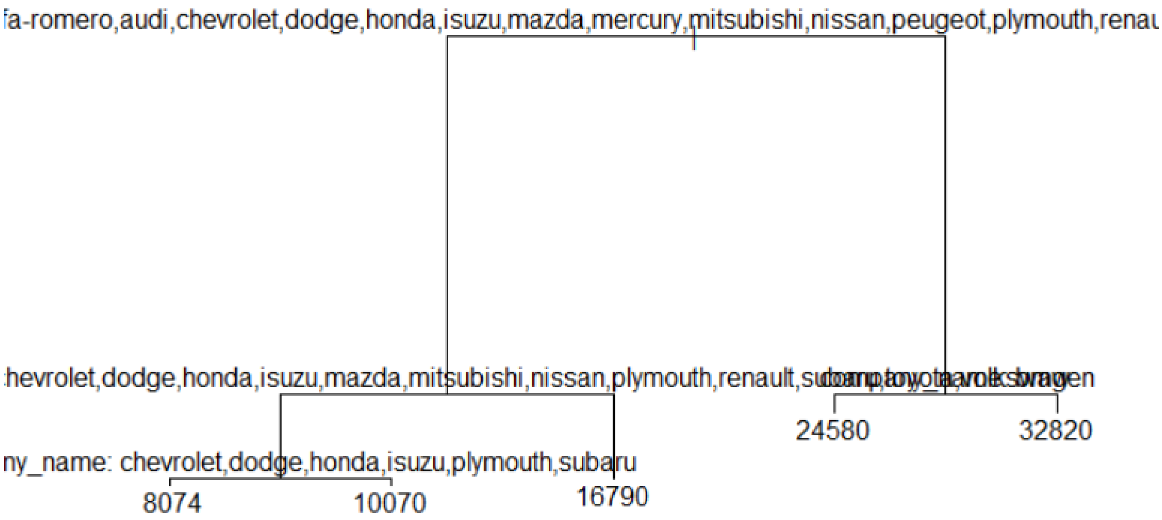
Tree Building

Decision tree is a graph to represent choices and their results in form of a tree. The nodes in the graph represent an event or choice and the edges of the graph represent the decision rules or conditions. It is mostly used in Machine Learning in R.

````
```{r}
tree_fit=tree(price~company_name,data=train)
tree_fit
```
````

```
node), split, n, deviance, yval
      * denotes terminal node

1) root 184 1.058e+10 12960
  2) company_name: alfa-
romero,audi,chevrolet,dodge,honda,isuzu,mazda,mercury,mitsubishi,nissan,peugeot,plymouth,renault,saab,subaru,toyota,volkswagen
 166 3.124e+09 11030
    4) company_name: chevrolet,dodge,honda,isuzu,mazda,mitsubishi,nissan,plymouth,renault,subaru,toyota,volkswagen 130
1.249e+09  9398 *
    5) company_name: alfa-romero,audi,mercury,peugeot,saab,volvo 36 2.834e+08 16910 *
  3) company_name: bmw,buick,jaguar,porsche 18 1.090e+09 30830
    6) company_name: bmw 7 5.980e+08 26340 *
    7) company_name: buick,jaguar,porsche 11 2.608e+08 33690 *
```



- The average price of dodge,Honda,isuzu,mazdawill be 9328
- The average price of nissan,Plymouth,Mitsubishi will be 16630
- The average price of Renault,Subaru,toyato,saab will be 26120
- The average price of Volkswagen,BMW will be 33120
- Totally it consider 4 leaf nodes from the root node.

```
```{r}
tr_predict=predict(tree_fit,newdata=test)
tr_predict
```
```

```
      15       22       32       48       50       57       59       70       74       77       95      105      112 
26340.71  9398.10  9398.10 33688.59 33688.59  9398.10  9398.10 33688.59 33688.59  9398.10  9398.10  9398.10 16911.87 
     126      133      134      152      164      168      176      191 
33688.59 16911.87 16911.87  9398.10  9398.10  9398.10  9398.10  9398.10 
```

```
```{r}
mean((tr_predict-test$price)^2)
```
```

```
[1] 20957609
```

According to decision tree error rate will be **20957609**

Random Forest

- Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.
- Random Forest can be used to solve regression and classification problems. In regression problems, the dependent variable is continuous.
- Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees. This reduces the variance when we average the trees.
- As in bagging, we build a number of decision trees on bootstrapped training samples.
- But when building these decision trees, each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of p predictors.

```
## Random Forest :

```{r}
set.seed(1)
rfm=randomForest(train$price ~company_name,data=train)
summary(rfm)
```
```

```
                Length Class  Mode
call              3     -none- call
type              1     -none- character
predicted       184     -none- numeric
mse             500     -none- numeric
rsq             500     -none- numeric
oob.times       184     -none- numeric
importance        1     -none- numeric
importanceSD      0     -none- NULL
localImportance   0     -none- NULL
proximity         0     -none- NULL
ntree             1     -none- numeric
mtry              1     -none- numeric
forest           11     -none- list
coefs             0     -none- NULL
y               184     -none- numeric
test              0     -none- NULL
inbag             0     -none- NULL
terms             3     terms  call
```

```
```{r}
predict_rfm= predict(rfm, newdata=test)
predict_rfm
```
```

```
       15        22        32        48        50        57        59        70        74        77        95       105 
26358.303  8163.440  8319.117 25902.310 25902.310 10206.811 10206.811 33361.707 33361.707  9544.846 10083.123 10083.123 
      112       126       133       134       152       164       168       176       191 
15461.030 33370.350 16760.733 16760.733 10155.582 10155.582 10155.582 10155.582 10128.217 
```

```
```{r}
mean((predict_rfm-test$price)^2)
```
```

```
[1] 25851872
```

According to random forest error rate will be **25851872**

CHAPTER V

PERFORMANCE EVALUATION

- Evaluating machine learning algorithm is an essential part of any project. The model may give satisfying results when evaluated using a metric accuracy score but may give poor results when evaluated against other metrics such as logarithmic loss or any other such metric.

- The performance measure is the way to evaluate a solution to the problem. It is the measurement that will make of the predictions made by a trained model on the test dataset. Performance measures are typically specialized to the class of problem that are working with, for example classification, regression, and clustering. Many standard performance measures will give a score that is meaningful to the problem domain.

- For example, classification accuracy for classification (total correct correction divided by the total predictions made multiple by 100 to turn it into a percentage).

- Since this project is related to regression model, the commonly used performance measure is mean squared error (MSE). In statistics, the mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors that is, the average squared difference between the estimated values and what is estimated. MSE is a risk function, corresponding to the expected value of the squared error loss. The fact that MSE is almost always strictly positive (and not zero) is because of randomness or because the estimator does not account for information that could produce a more accurate estimate.

MSE:

Mean squared error is an estimator measures the average of the squares of the errors that is the average squared difference between the estimated value and actual value.

MSE of Linear Regression

```
```{r}
mean((tr-test$price)^2)
```

[1] 19999973
```

According to decision tree error rate will be **20957609**

Accuracy rate will be **81%**

MSE of Decision Tree

```{r}
mean((tr_predict-test$price)^2)
```

```
[1] 20957609
```

According to decision tree error rate will be **20957609**

Accuracy rate will be **80%**

MSE of Random Forest

```{r}
mean((predict_rfm-test$price)^2)
```

```
[1] 25851872
```

According to random forest error rate will be **25851872**

Accuracy rate will be **75%**

CHAPTER VI

CONCLUSION

- The automotive industry in the United States began in the 1890s and, as a result of the size of the domestic market and the use of mass production, rapidly evolved into the largest in the world. However, the United States was overtaken by Japan as the largest automobile producer in the 1980s, and subsequently by China in 2008.

- American manufacturers produce approximately 8–10 million units annually. Notable exceptions were 5.7 million automobiles manufactured in 2009 (due to crisis), while production peaked during the 1970s and early 2000s at levels of 13–15 million units

- A multivariate regression based solution is proposed to calculate selling price of each car available in the U.S.Market.

- The U.S. is currently second among the largest manufacturer(s) in the world by volume.

- According to our prediction random forest tree is the best method to predict cars selling price.

REFERENCES

[1]V. Lazarov and M. Capota. Churn Prediction. Business Analytics Course. TUM Computer Science, December 2007. http://home.in.tum.de/~lazarov/files/research/papers/churn- prediction.pdf

[2] Carlo Vercellis, Business Intelligence: Data Mining and Optimization for Decision Making, John Wiley & Sons, Ltd. 2009 ISBN: 978-0-470- 51138-1

[3] S. Gotovac. “Modeling Data Mining Applications for Prediction of Prepaid Churn in Telecommunication Services,” vol. 51, no. 3, pp. 275-283, 2010

[4] H. Kim, and C. Yoon, “Determinants of subscriber churn and customer loyalty in the Korean mobile telephony market.” Telecommunications Policy. Vol. 28 No.: PP. 751-765, 2004.

[5]W. Au, C. Chan, and X. Yao, A Novel Evolutionary Data Mining Algorithm with Applications to Churn Prediction, IEEE transactions on evolutionary computation, 7, 6, 532-545,2003.

[6] R. Behara, W. Fisher, and J. Lemmink, Modelling and Evaluating Service Quality Measurement Using Neural Networks, International journal of operations and production management, 22, 10, 1162-1185, 2002.