SUJAL VERMA

2023

DATA ANALYST PORTFOLIO

Professional Background

Highly motivated and diligent individual with a passion for data analysis and a strong foundation in economics. My name is Sujal Verma, and I am currently in the final year of pursuing my Bachelor of Arts in Economics. Throughout my academic journey, I have developed a comprehensive understanding of economic principles and their application in real-world scenarios.

In addition to my academic pursuits, I have honed my skills in various data-related tools and technologies. Proficient in SQL, Microsoft Excel, Python, Tableau, and Google Sheets, I possess the technical prowess to efficiently collect, clean, and analyse data. My expertise in data analysis allows me to derive meaningful insights, aiding in making informed decisions and solving complex problems.

As a recent graduate, I may have limited formal experience, but my passion for data analysis has driven me to explore practical projects and internships. These experiences have provided me with invaluable hands-on exposure and have strengthened my ability to work collaboratively within diverse teams.

In every endeavour, I approach tasks with a meticulous eye for detail, ensuring accuracy and precision in my work. My eagerness to learn and adapt to new challenges enables me to stay up-to-date with emerging trends in the field of data analysis.

With a solid academic background and a versatile skill set, I am poised to make a meaningful contribution to the world of data analysis. I am actively seeking opportunities to apply my knowledge and skills in a dynamic and innovative environment where I can contribute to data-driven decision-making and contribute to the success of an organization.

Thank you for considering my profile, and I look forward to discussing how my capabilities can align with your organization's objectives.

Table Of Contents

Data Analytics Process

Description

We use Data Analytics in everyday life without even knowing it.
For e.g. : Going to a market to buy something .

Your task is to give the example(s) of such a real-life situation where we use Data Analytics and link it with the data analytics process.

Ex 1 - Buying a mobile/laptop.

1. **Plan** – we first decide our usage like do we need it for ?.
Ex- photography, Gaming, general purposes etc.
2. **Prepare** – we will decide a certain amount/budget to spend of the device. Lets say 60000/- INR
3. **Process** – we will choose a (brand) out of the other brands offering the product. Ex Samsung or HP.
4. **Analyse** – now we will analyse the best suitable model with other models offered by the same brand. Ex inspiron or vostro in case of dell.
5. **Share** – now we will share it with the friends or shopkeeper to give their suggestions
6. **Act** – we will buy the best fit device.

Instagram User Analytics

Description

User analysis involves tracking how users engage with a digital product, such as a software application or a mobile app. The insights derived from this analysis can be used by various teams within the business. For example, the marketing team might use these insights to launch a new campaign, the product team might use them to decide on new features to build, and the development team might use them to improve the overall user experience. In this project, you'll be using SQL and MySQL Workbench as your tool to analyze Instagram user data and answer questions posed by the management team. Your insights will help the product manager and the rest of the team make informed decisions about the future direction of the Instagram app. Remember, the goal of this project is to use your SQL skills to extract meaningful insights from the data. Your findings could potentially influence the future development of one of the world's most popular social media platforms.

SQL Tasks

A) Marketing Analysis:

Loyal User Reward: The marketing team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.
Your Task: Identify the five oldest users on Instagram from the provided database.

Inactive User Engagement: The team wants to encourage inactive users to start posting by sending them promotional emails.
Your Task: Identify users who have never posted a single photo on Instagram.

Contest Winner Declaration: The team has organized a contest where the user with the most likes on a single photo wins.
Your Task: Determine the winner of the contest and provide their details to the team.

Hashtag Research: A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.
Your Task: Identify and suggest the top five most commonly used hashtags on the platform.

Ad Campaign Launch: The team wants to know the best day of the week to launch ads.
Your Task: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

SQL Tasks :

B) Investor Metrics:

•**User Engagement:** Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.

•**Your Task: Calculate the average number of posts per user on Instagram. Also, provide the total number of photos on Instagram divided by the total number of users**.

•**Bots & Fake Accounts:** Investors want to know if the platform is crowded with fake and dummy accounts.

•**Your Task: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user**.

Design

Using the 'create db' function of MySQL create a data base Then add tables and column names Then add the values into them using the 'insert into' function of MySQL

By using the 'select' command we can query the desired output Steps taken to load the data into the data base Software used for querying the results --> MySQL Workbench 8.0 CE

Q.1 Rewarding the most Loyal users:

People who have been using the platform for the longest time.(Top 5 oldest Instagram users)

To find the most loyal i.e. the top 5 oldest users of Instagram:
1. We will use the data from the users table by selecting the username and created_at columns.
2. Then using the order by function we will order the desired output by sorting with the created_at column in ascending order.
3. Then using the limit function, the output will be displayed for top 5 oldest Instagram users

Program/Query:

```
select username, created_at
from users
order by created_at ASC
limit 5;
```

Result :

username	created_at
Darby_Herzog	06-05-2016 00:14
Emilio_Bernier52	06-05-2016 13:04
Elenor88	08-05-2016 01:30
Nicole71	09-05-2016 17:30
Jordyn.Jacobson2	14-05-2016 07:56

Q.2 Remind Inactive Users to Start Posting : Remind Inactive users to Start Posting(Users who never posted a single photo on Instagram)

To Find the most inactive users i.e. the users who have never posted a single photo on Instagram:

1. We will first select username column from the users table.
2. Then we will left join photos table on the users table, on users.id =
photos.user_id because, both the users.id and photos.user_id have
common contents in them.
3. Then we will find rows from the users table where the photos.id IS
NULL

Program/Query:

```
select username, users.id as
user_id
from users
left join photos
on users.id = photos.user_id
where photos.id IS NULL
order by users.id;
```

Result

username	user_id
Aniya_Hackett	5
Kasandra_Homenick	7
Jaclyn81	14
Rocio33	21
Maxwell.Halvorson	24
Tierra.Trantow	25
Pearl7	34
Ollie_Ledner37	36
Mckenna17	41
David.Osinski47	45
Morgan.Kassulke	49
Linnea59	53
Duane60	54
Julien_Schmidt	57
Mike.Auer39	66
Franco_Keebler64	68
Nia_Haag	71
Hulda.Macejkovic	74
Leslie67	75
Janelle.Nikolaus81	76
Darby_Herzog	80
Esther.Zulauf61	81
Bartholome.Bernhard	83
Jessyca_West	89
Esmeralda.Mraz57	90
Bethany20	91

So, there are in total 26 users of
the 100 users who have never posted a single photo on Instagram

Q.3 Declaring Contest Winner : The team started a contest and the user who gets the most likes on a single photo will win the contest now they wish to declare the winner. Identify the winner of the contest and provide their details to the team.

To find the most the username, photo_id, image_url and total_number_of_likes of that image:

1. First we will select the users.username, photos.id, photos.image_url and
count(*) as total
2. Then, we will inner join the three tables wiz : photos, likes and users, on
likes.photo_id = photos.id and photos.user_id = users.id
3. Then, by using group by function we will group the output on the basis
of photos.id
4. Then, using order by function we will sorting the data on the basis of
the total in descending order
5. Then, to find the most liked photo we will using limit function to view
only the top liked photo's information

Program/Query :

```
select users.id as user_id, users.username, photos.id as
photo_id,
photos.image_url, count(*) as total
from photos
inner join likes
on likes.photo_id = photos.id
inner join users
on photos.user_id = users.id
group by photos.id
order by total DESC
limit 1;
```

Result

user_id	username	photo_id	image_url	total
52	Zack_Kemmer93	145	https://jarret.name	48

So, the user named Zack_Kemmer93 with user_id 52 is the winner of the contest cause his photo with photo_id 145 has the
highest number of likes i.e. 48

Q.4 Hashtag Researching : A partner brand wants to know, which hashtags to use in the post to reach the most people on the platform.(Top 5 commonly used #Hashtags on Instagram)

To find the top 5 most commonly used hashtags on Instagram:
1. We need to select the tag_name column from the tag table and the
count(*) as total function so as to count the number of tags used
individually.
2. Then, we need to join tags table and photo_tags table, on tags.id =
photo_tags.tag_id cause they contain the same content in them i.e.
tag_id
3. Then using the group by function we need to group the desired output
on the basis of tags.tag_name
4. Then using the order by function we need to sort the output on the
basis of total(total number of tags per tag_name) in descending order
5. Then, to find the top 5 most used tag names we will use the limit 5
function

Program/Query :

```
select tags.tag_name, count(*) as
total_number_of_times_tag_used_individually
from tags
join photo_tags
on tags.id = photo_tags.tag_id
group by tags.tag_name
order by total_number_of_times_tag_used_individually DESC
limit 5
```

Result

tag_name	total_number_of_times_tag_used_individually
smile	59
beach	42
party	39
fun	38
concert	24

Q.5 Launch AD Campaign : The team wants to know, which day would be the best day to launch ADs. (What day of the week do most users register on?)

To find the day of week on which most users register on Instagram:

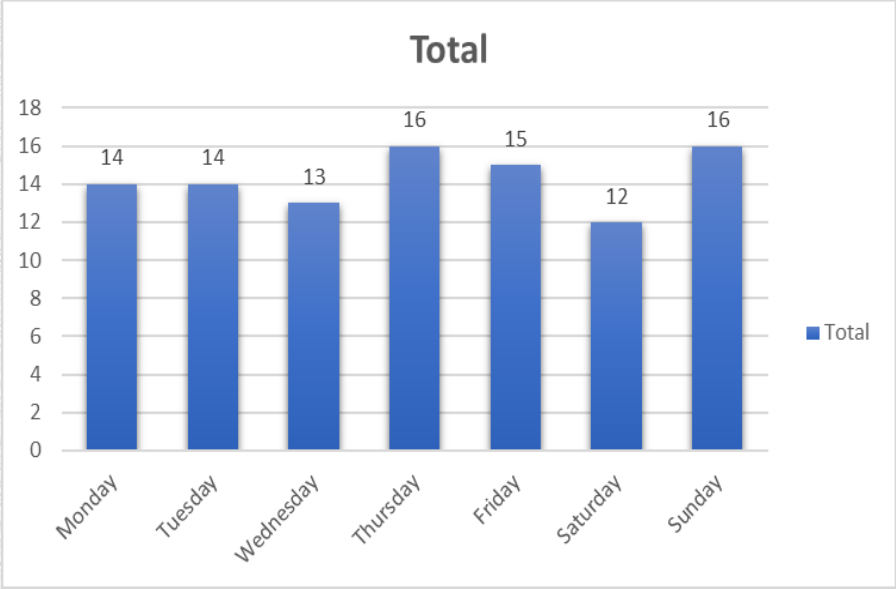
1. First we define the columns of the desired output table using select dayname(created_at) as day_of_week and count(*) as total_number_of_users_registered from the users table
2. Then using the group by function we group the output table on the basis of day_of_week
3. Then using the order by function we order/sort the output table on the basis of total_number_of_users_registered in descending order

Program/Query :

```
select dayname(created_at) as day_of_week,
count(*) as
total_number_of_users_registered
from users
group by day_of_week
order by total_number_of_users_registered
DESC;
```

Result

day_of_week	total_number_of_users_registered
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12



~ Most of the users registered on Thursday and Sunday i.e. 16 and hence it would prove beneficial to start AD Campaign on these two days

Investor Metrics

Q.1 User Engagement : Are users still as active and post on Instagram or
they are making fewer posts.
How many times does average user posts on Instagram?
Also, provide the total number of photos on Instagram/total number of users.

To find the how many times does average posts on Instagram:
1. First, we need to find first the count number of photos (posts) that are present in the photos.id column of the photos table
i.e. count(*) from photos

2. Similarly, we need to find the number of users that are present in the users.id column of the users table i.e. count(*) from users

3. Next, we need to divide both the values i.e. count(*) from photos/count(*) from users and hence we would get the total number of photos / total number of users

4. To find how many times the users posts on Instagram we need to find
the total occurrences of each user_id in photos table

Program/Query to find (total number of photos/total number of users) :

```
select
(select count(*) from photos)/(select count(*)
from users) as
total_photos_divide_total_photos;
```

Result

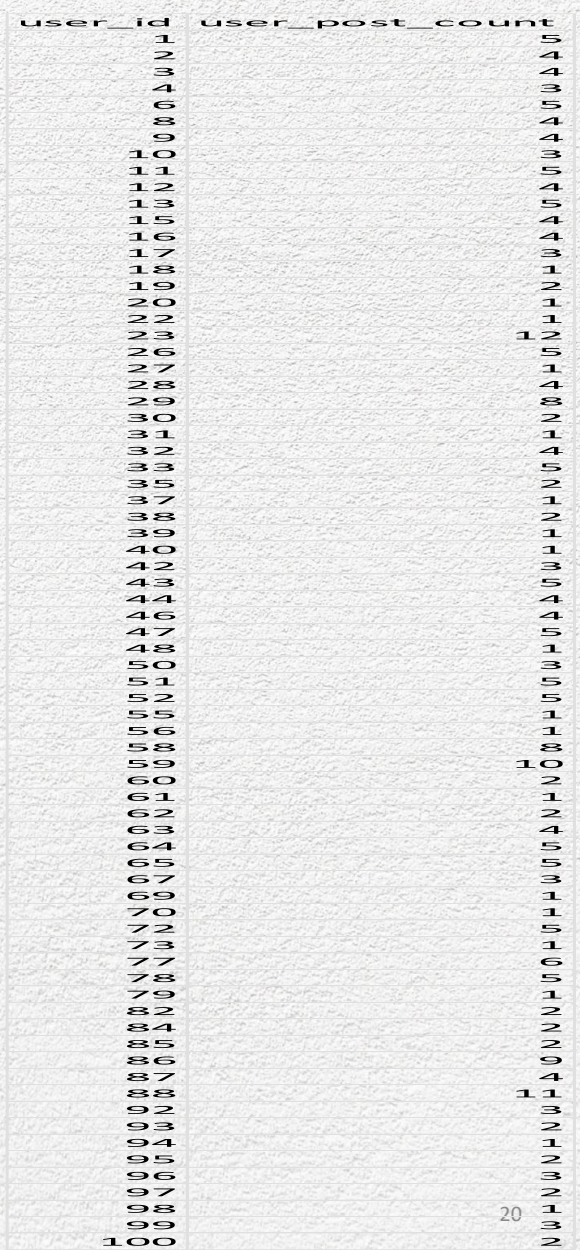
total_photos_divide_total_photos
2.57

So, there are in total 257 rows i.e. 257 photos in the photos table and
100 rows i.e. 100 ids in the users table which makes the desired output
to be 257/100 = 2.57

Program/Query to find the times each user posts on Instagram :

```
select user_id,count(*) as user_post_count
from photos
group by user_id
order by user_id;
```

Result



user_id	user_post_count
1	5
2	4
3	4
4	3
5	4
6	4
8	5
9	4
10	3
11	5
12	4
13	5
14	4
15	4
16	4
17	3
18	1
19	2
20	1
22	1
23	2
24	1
26	5
27	1
28	4
29	8
30	2
31	1
32	4
33	5
35	2
37	1
38	2
39	1
40	1
42	3
43	5
44	4
46	4
47	5
48	1
50	3
51	5
52	5
55	1
56	1
58	1
59	8
60	2
61	1
62	2
63	4
64	5
65	5
67	3
69	1
70	1
72	5
73	1
77	6
78	5
79	1
82	2
84	2
85	2
86	9
87	4
88	1
92	3
93	2
94	1
95	2
96	3
97	2
98	1
99	3
100	2

So the user_id along with the number of times each user_id has posted is provided

Q.2 Bots and Fake Accounts : The investors want to know if the platform is crowded with fake and dummy accounts. Provide data on users (bots) who have liked every single photo on the site (since any normal user would not be able to do this).

To find the bots and fake accounts :
1. First, we select the user_id column from the photos table
2. Then we select the username column from the users table
3. Then, we select the count(*) function to count total number of likes
from the likes table
4. Then we inner join users and likes table on the basis of users.id and
likes.user_id, using the on function/clause
5. Then by using the group by function we group the desired output
table on the basis of likes.user_id
6. Then, we search for the values from the cout(*) from photos having
equal values with the total_likes_per_user

Program/Query :

```
select user_id, username, count(*) as total_likes_per_user
from users
inner join likes
on users.id = likes.user_id
group by likes.user_id
having total_likes_per_user = (select count(*) from photos);
```

Result

user_id	username	total_likes_per_user
5	Aniya_Hackett	257
14	Jaclyn81	257
21	Rocio33	257
24	Maxwell.Halvorson	257
36	Ollie_Ledner37	257
41	Mckenna17	257
54	Duane60	257
57	Julien_Schmidt	257
66	Mike.Auer39	257
71	Nia_Haag	257
75	Leslie67	257
76	Janelle.Nikolaus81	257
91	Bethany20	257

So, the users along with their respective username, user_id and total_likes_per_user have been provided. This user_ids may be bots or fake accounts

Operations & Metric Analytics

Description

Operational Analytics is a crucial process that involves analyzing a company's end-to-end operations. This analysis helps identify areas for improvement within the company. As a Data Analyst, you'll work closely with various teams, such as operations, support, and marketing, helping them derive valuable insights from the data they collect.

One of the key aspects of Operational Analytics is investigating metric spikes. This involves understanding and explaining sudden changes in key metrics, such as a dip in daily user engagement or a drop in sales. As a Data Analyst, you'll need to answer these questions daily, making it crucial to understand how to investigate these metric spikes.

Analysis done on the following points:-

Case Study 1 : Job Data

A. Number of jobs reviewed: Amount of jobs reviewed over time.
Your task: Calculate the number of jobs reviewed per hour per day for November 2020?

B. Throughput: It is the no. of events happening per second.
Your task: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput?
For throughput, do you prefer daily metric or 7-day rolling and why?

C. Percentage share of each language: Share of each language for different contents.
Your task: Calculate the percentage share of each language in the last 30 days?

D. Duplicate rows: Rows that have the same value present in them.
Your task: Let's say you see some duplicate rows in the data. How will you
display duplicates from the table?

Software used : MySQL Workbench 8.0 CE

Case Study 2: Investigating metric spike

A. User Engagement: To measure the activeness of a user. Measuring if the user
finds quality in a product/service.
Your task: Calculate the weekly user engagement?

B. User Growth: Amount of users growing over time for a product.
Your task: Calculate the user growth for product?

C. Weekly Retention: Users getting retained weekly after signing-up for a product.
Your task: Calculate the weekly retention of users-sign up cohort?

D. Weekly Engagement: To measure the activeness of a user. Measuring if the user
finds quality in a product/service weekly.
Your task: Calculate the weekly engagement per device?

E. Email Engagement: Users engaging with the email service.
Your task: Calculate the email engagement metrics?

Software Used : MySQL Workbench 8.0 CE

Job Data

Number of jobs reviewed: Amount of jobs reviewed over time.
Calculate the number of jobs reviewed per hour per day for November 2020?

To find the number of jobs reviewed per hour per day of November 2020:

1. We will use the data from job_id columns of the job_data table.

2. Then we will divide the total count of job_id (distinct and non-distinct) by (30 days * 24 hours)for finding the number of jobs reviewed per day

Program/Query (non_distinct_job_id):

```
select
count(job_id)/(30*24) as
number_of_jobs_reviewed_per_day_non_distinct
from job_data;
```

Result

number_of_jobs_reviewed_per_day_non_distinct
0.0111

Number of jobs reviewed: Amount of jobs reviewed over time. Calculate the number of jobs reviewed per hour per day for November 2020?

Program/Query (distinct_job_id):

```
select
count(distinct job_id)/(30*24) as
number_of_jobs_reviewed_per_day_distinct
from job_data;
```

Result

number_of_jobs_reviewed_per_day_distinct
0.0083

Job Data

Throughput: It is the no. of events happening per second.

Let's say the above metric is called throughput. Calculate 7 day rolling
average of throughput? For throughput, do you prefer daily metric or 7-day
rolling and why?

For calculating the throughput we will be using the 7-day rolling because 7-day rolling gives us the average for all the days right from day 1 to day 7 Whereas daily metric gives us average for only that particular day itself.

For calculating the 7-day rolling daily metric average of throughput:-

1. We will be first taking the count of job_id(distinct and non-distinct) and ordering them w.r.t ds (date of interview)

2. Then by using the ROW function we will be considering the rows between 6 preceding rows and the current row

3. Then we will be taking the average of the jobs_reviewed

Program/Query (distinct_job_id):

```
SELECT ds as date_of_review, jobs_reviewed,
AVG(jobs_reviewed)
OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND
CURRENT ROW) AS
throughput_7_rolling_average
FROM
(
SELECT ds, COUNT( DISTINCT job_id) AS jobs_reviewed
FROM job_data
GROUP BY ds ORDER BY ds
) a;
```

Result

date_of_review	jobs_reviewed	throughput_7_rolling_average
25-11-2020	1	1
26-11-2020	1	1
27-11-2020	1	1
28-11-2020	2	1.25
29-11-2020	1	1.2
30-11-2020	2	1.3333

Program/Query (non_distinct_job_id):

```
SELECT ds as date_of_review, jobs_reviewed,
AVG(jobs_reviewed)
OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND
CURRENT ROW) AS
throughput_7_rolling_average_non_distinct_job_id
FROM
(
SELECT ds, COUNT(job_id) AS jobs_reviewed
FROM job_data
GROUP BY ds ORDER BY ds
) a;
```

Result

date_of_review	jobs_reviewed	throughput_7_rolling_average_non_distinct_job_id
25-11-2020	1	1
26-11-2020	1	1
27-11-2020	1	1
28-11-2020	2	1.25
29-11-2020	1	1.2
30-11-2020	2	1.3333

Job Data

Percentage share of each language: Share of each language for different
contents.

Calculate the percentage share of each language?
To calculate the percentage share of each language (distinct and nondistinct):-

1. We will first divide the total number of languages (distinct/non-distinct) by the total number of rows presents in the table

2. Then we will do the grouping based on the languages.

Program/Query (non_distinct_language):

```
select
job_data.job_id, job_data.language,
count(job_data.language) as
total_of_each_language,
((count(job_data.language)/(select count(*) from
job_data))*100) as
percentage_share_of_each_language
from job_data
group by job_data.language;
```

Result

job_id	language	total_of_each_language	percentage_share_of_each_language
21	English	1	12.5
22	Arabic	1	12.5
23	Persian	3	37.5
25	Hindi	1	12.5
11	French	1	12.5
20	Italian	1	12.5

Program/Query (distinct_language):

```
select
job_data.job_id, job_data.language,
count(distinct job_data.language) as
total_of_each_language,
((count(job_data.language)/(select count(*) from
job_data))*100) as
percentage_share_of_each_distinct_language
from job_data
group by job_data.language;
```

Result

job_id	language	total_of_each_language	percentage_share_of_each_distinct_language
22	Arabic	1	12.5
21	English	1	12.5
11	French	1	12.5
25	Hindi	1	12.5
20	Italian	1	12.5
23	Persian	1	37.5

Job Data

Duplicate rows: Rows that have the same value present in them.
Let's say you see some duplicate rows in the data. How will you display duplicates
from the table?
To view the duplicate rows having the same value we will:-
1. First decide in which do we need to find the duplicate row values
2. After deciding the column(parameter) we will use the ROW_NUMBER
function to find the row numbers having the same value
3. Then we will portioning the ROW_NUMBER function over the column
(parameter) that we decided i.e. job_id
4. Then using the WHERE function we will find the row_num having value
greater than 1 i.e. row_num > 1 based on the occurrence of the job_id in
the table.

Program/Query :

```
SELECT *
FROM
(
SELECT *, ROW_NUMBER()OVER(PARTITION BY job_id) AS row_num
FROM job_data
) a
WHERE row_num>1;
```

Program/Query :

```
SELECT *
FROM
(
SELECT *, ROW_NUMBER()OVER(PARTITION BY
job_id) AS row_num
FROM job_data
) a
WHERE row_num>1;
```

Result

ds	job_id	actor_id	event	language	time_spent	org	row_num
28-11-2020	23	1005	transfer	Persian	22	D	2
26-11-2020	23	1004	skip	Persian	56	A	3

Investigating Metric Spike

User Engagement: To measure the activeness of a user. Measuring if the user finds
quality in a product/service.
Your task: Calculate the weekly user engagement?

To find the weekly user engagement:-

1. We will extract the week from the occurred_at column of the events table
using the EXTRACT function and WEEK function

2. Then we will be counting the number of distinct user_id from the events table

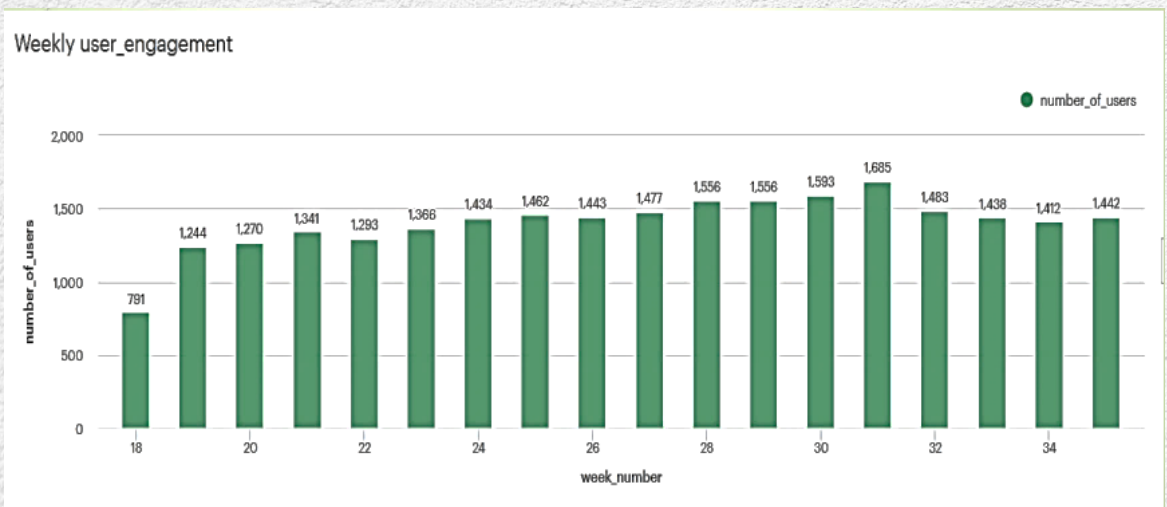
3. Then we will use the GROUP BY function to group the output w.r.t week from
occurred_at

Program/Query :

```
SELECT
extract (week from occurred_at) as week_number,
count(distinct user_id) as number_of_users
FROM
tutorial.yammer_events
group by
week_number;
```

Result

week_number	number_of_users
18	791
19	1244
20	1270
21	1341
22	1293
23	1366
24	1434
25	1462
26	1443
27	1477
28	1556
29	1556
30	1593
31	1685
32	1483
33	1438
34	1412
35	1442



Investigating Metric Spike

User Growth: Amount of users growing over time for a product.
Your task: Calculate the user growth for product?
User Growth = Number of active users per week

To find the user growth (number of active users per week):-
1. First we will the extract the year and week for the occurred_at column of the
users table using the extract, year and week functions
2. Then we will group the extracted week and year on the basis of year and week
number
3. Then we ordered the result on the basis of year and week number
4. Then we will find the cumm_active_users using the SUM, OVER and ROW
function between unbounded preceding and current row

Program/Query :

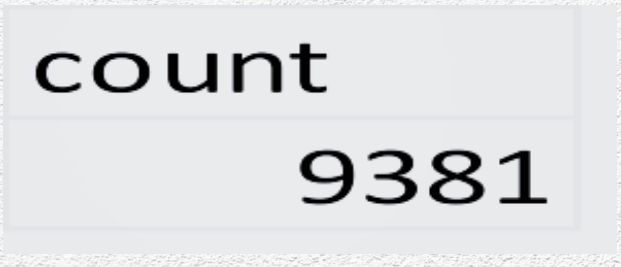
```
select
year_num,
week_num,
num_active_users,
SUM(num_active_users)OVER(ORDER BY year_num,
week_num ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW) AS
cum_active_users
from
(
select
extract (year from a.activated_at) as year_num,
extract (week from a.activated_at) as week_num,
count(distinct user_id) as num_active_users
from
tutorial.yammer_users a
WHERE
state = 'active'
group by year_num,week_num
order by year_num,week_num
) a;
```

Result

year_num	week_num	num_active_users	cum_active_users	year_num	week_num	num_active_users	cum_active_users
2013	1	67	67	2013	45	97	2564
2013	2	29	96	2013	46	94	2658
2013	3	47	143	2013	47	82	2740
2013	4	36	179	2013	48	103	2843
2013	5	30	209	2013	49	96	2939
2013	6	48	257	2013	50	117	3056
2013	7	41	298	2013	51	123	3179
2013	8	39	337	2013	52	104	3283
2013	9	33	370	2014	1	91	3374
2013	10	43	413	2014	2	122	3496
2013	11	33	446	2014	3	112	3608
2013	12	32	478	2014	4	113	3721
2013	13	33	511	2014	5	130	3851
2013	14	40	551	2014	6	132	3983
2013	15	35	586	2014	7	135	4118
2013	16	42	628	2014	8	127	4245
2013	17	48	676	2014	9	127	4372
2013	18	48	724	2014	10	135	4507
2013	19	45	769	2014	11	152	4659
2013	20	55	824	2014	12	132	4791
2013	21	41	865	2014	13	151	4942
2013	22	49	914	2014	14	161	5103
2013	23	51	965	2014	15	166	5269
2013	24	51	1016	2014	16	165	5434
2013	25	46	1062	2014	17	176	5610
2013	26	57	1119	2014	18	172	5782
2013	27	57	1176	2014	19	160	5942
2013	28	52	1228	2014	20	186	6128
2013	29	71	1299	2014	21	177	6305
2013	30	66	1365	2014	22	186	6491
2013	31	69	1434	2014	23	197	6688
2013	32	66	1500	2014	24	198	6886
2013	33	73	1573	2014	25	222	7108
2013	34	70	1643	2014	26	210	7318
2013	35	80	1723	2014	27	199	7517
2013	36	65	1788	2014	28	223	7740
2013	37	71	1859	2014	29	215	7955
2013	38	84	1943	2014	30	228	8183
2013	39	92	2035	2014	31	234	8417
2013	40	81	2116	2014	32	189	8606
2013	41	88	2204	2014	33	250	8856
2013	42	74	2278	2014	34	259	9115
2013	43	97	2375	2014	35	266	9381
2013	44	92	2467				

Program/Query : select count(*) Program/Query : select count(*) from tutorial.yammer_users where state = 'active';from tutorial.yammer_users where state = 'active';

Result



count
9381

Hence, there are in total 9381 active users from 1st week of 2013 to the 35th week of 2014

Investigating Metric Spike

Weekly Retention: Users getting retained weekly after signing-up for a product.
Your task: Calculate the weekly retention of users-sign up cohort?
The weekly retention of users-sign up cohort can be calculated by two means i.e.
either by specifying the week number (18 to 35) or for the entire column of
occurred_at of the events table.
1. Firstly we will extract the week from occurred_at column using the extract,week
functions
2. Then, we will select out those rows in which event_type = 'signup_flow' and
event_name = 'complete_signup'
3. If finding for a spectifc week we will spectify the week number using the extract
function
4. Then using the left join we will join the two tables on the basis of user_id where
event_type = 'engagement'
5. Then we will use the Group By function to group the output table on the basis of
user_id
6. Then we will use the Order By function to order the result table on the basis of
user_id

Program/Query(Without Specifying the week number) :

```
SELECT
distinct user_id,
COUNT(user_id),
SUM(CASE WHEN retention_week = 1 Then 1 Else 0 END) as per_week_retention
FROM
(
SELECT
a.user_id,
a.signup_week,
b.engagement_week,
b.engagement_week - a.signup_week as retention_week
FROM
(
(SELECT distinct user_id, extract(week from occurred_at) as signup_week from tutorial.yammer_events
WHERE event_type = 'signup_flow'
and event_name = 'complete_signup'
)a
LEFT JOIN
(SELECT distinct user_id, extract (week from occurred_at) as engagement_week FROM tutorial.yammer_events
where event_type = 'engagement'
)b
on a.user_id = b.user_id
)
)d
group by user_id
order by user_id;
```

Link for the saved result –
https://drive.google.com/file/d/1EPbZJPYJxJCfVZqAo6NHn-bZTOD15IvQ/view

Program/Query(Specifying the week number as 18) :

```
SELECT
distinct user_id,
COUNT(user_id),
SUM(CASE WHEN retention_week = 1 Then 1 Else 0
END) as per_week_retention
FROM
(
SELECT
a.user_id,
a.signup_week,
b.engagement_week,
b.engagement_week - a.signup_week as
retention_week
FROM
(
(SELECT distinct user_id, extract(week from
occurred_at) as signup_week from
tutorial.yammer_events
WHERE event_type = 'signup_flow'
and event_name = 'complete_signup’
and extract(week from occurred_at) = 18
)a
LEFT JOIN
(SELECT distinct user_id, extract (week from
occurred_at) as engagement_week FROM
tutorial.yammer_events
where event_type = 'engagement'
)b
on a.user_id = b.user_id
)
)d
group by user_id
order by user_id;
```

Link for the saved result –
https://drive.google.com/file/d/1ktpgBTtufzPO2bYAHFEu5wbVJOGz1esA/view

Investigating Metric Spike

Weekly Engagement: To measure the activeness of a user. Measuring if the user finds
quality in a product/service weekly.
Your task: Calculate the weekly engagement per device?
To find the weekly user engagement per device:-
1. Firstly we will extract the year_num and week_num from the occurred_at column
of the events table using the extract, year and week function
2. Then we will select those rows where event_type = 'engagement' using the WHERE
clause
3. Then by using the Group By and Order By function we will group and order the
result on the basis of year_num, week_num and device

Program/Query :

```
SELECT
extract(year from occurred_at) as year_num,
extract(week from occurred_at) as week_num,
device,
COUNT(distinct user_id) as no_of_users
FROM
tutorial.yammer_events
where event_type = 'engagement'
GROUP by 1,2,3
order by 1,2,3;
```

Link for the saved result –
https://drive.google.com/file/d/1nVYPm4RsptUFqauyKA02-HTC6z8ALBXz/view

Investigating Metric Spike

Email Engagement: Users engaging with the email service.
Your task: Calculate the email engagement metrics?

To find the email engagement metrics(rate) of users:-
1. We will first categorize the action on the basis of email_sent, email_opened
and email_clicked using the CASE, WHEN, THEN functions
2. Then we select the sum of category of email_opened divide by the sum of
the category of email_sent and multiply the result by 100.0 and name is as
email_opening_rate
3. Then we select the sum of category of email_clicked divide by the sum of
the category of email_sent and multiply the result by 100.0 and name is as
email_clicking_rate
4. email_sent = ('sent_weekly_digest','sent_reengagement_email’)
5. email_opened = 'email_open’
6. email_clicked = 'email_clickthrough'

Program/Query :

```
SELECT
100.0*SUM(CASE when email_cat = 'email_opened'
then 1 else 0 end)/SUM(CASE when
email_cat = 'email_sent' then 1 else 0 end) as
email_opening_rate,
100.0*SUM(CASE when email_cat = 'email_clicked'
then 1 else 0 end)/SUM(CASE when
email_cat = 'email_sent' then 1 else 0 end) as
email_clicking_rate
FROM
(
SELECT
*,
CASE
WHEN action in
('sent_weekly_digest','sent_reengagement_email')
then 'email_sent'
WHEN action in ('email_open')
then 'email_opened'
WHEN action in ('email_clickthrough')
then 'email_clicked'
end as email_cat
from tutorial.yammer_emails
) a;
```

Link for the saved result –
https://drive.google.com/file/d/1z6FNGmuMe3i4VlZooGJCT1dYeZXiw6Hc/view

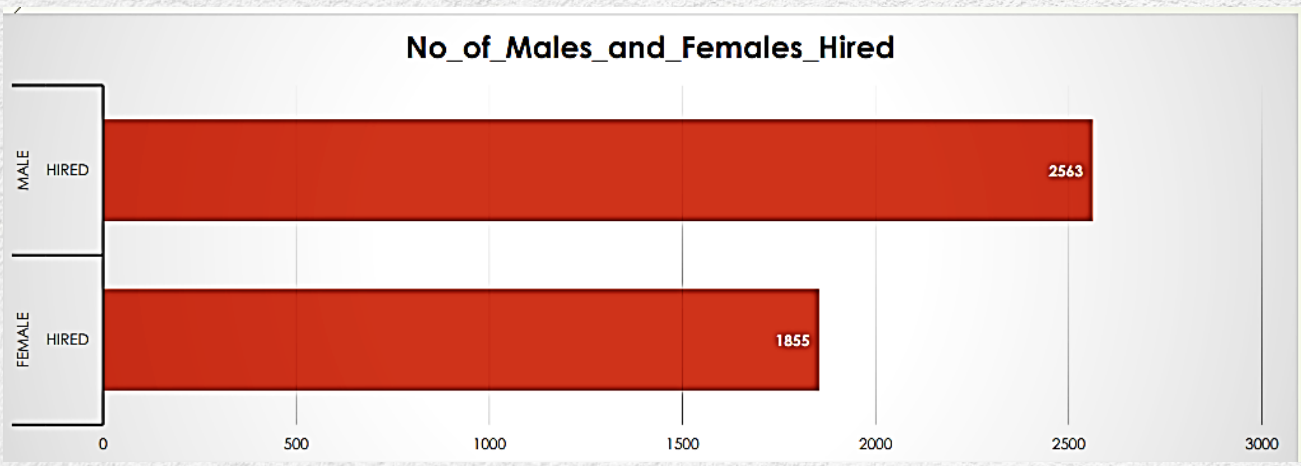
Hiring Process Analytics

Description

The hiring process is a crucial function of any company, and understanding trends such as the number of rejections, interviews, job types, and vacancies can provide valuable insights for the hiring department.

Q.1 How many males and females are Hired ?

event_name	Status	No_of_males_and_females_hired
Female	Hired	1855
Male	Hired	2563



Q.2 What is the average salary offered in this company ?

To find the average salary offered in this company:-

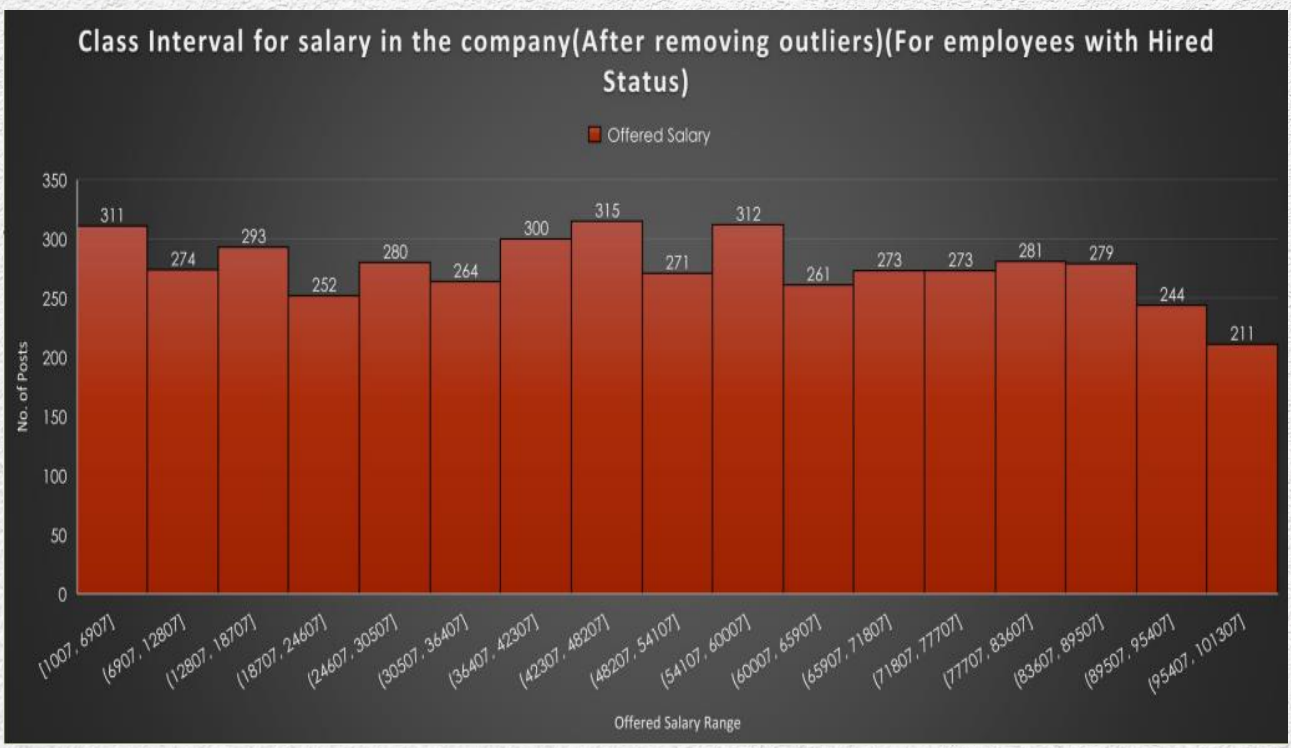
1. First, we need to remove the outliers
 i.e. to remove the salaries below 1000 and above 100000

2. Then using the formula
=AVERAGE(entire_column_of_salary_after_removing_outliers)

Formula Used:- =AVERAGE(G:G)

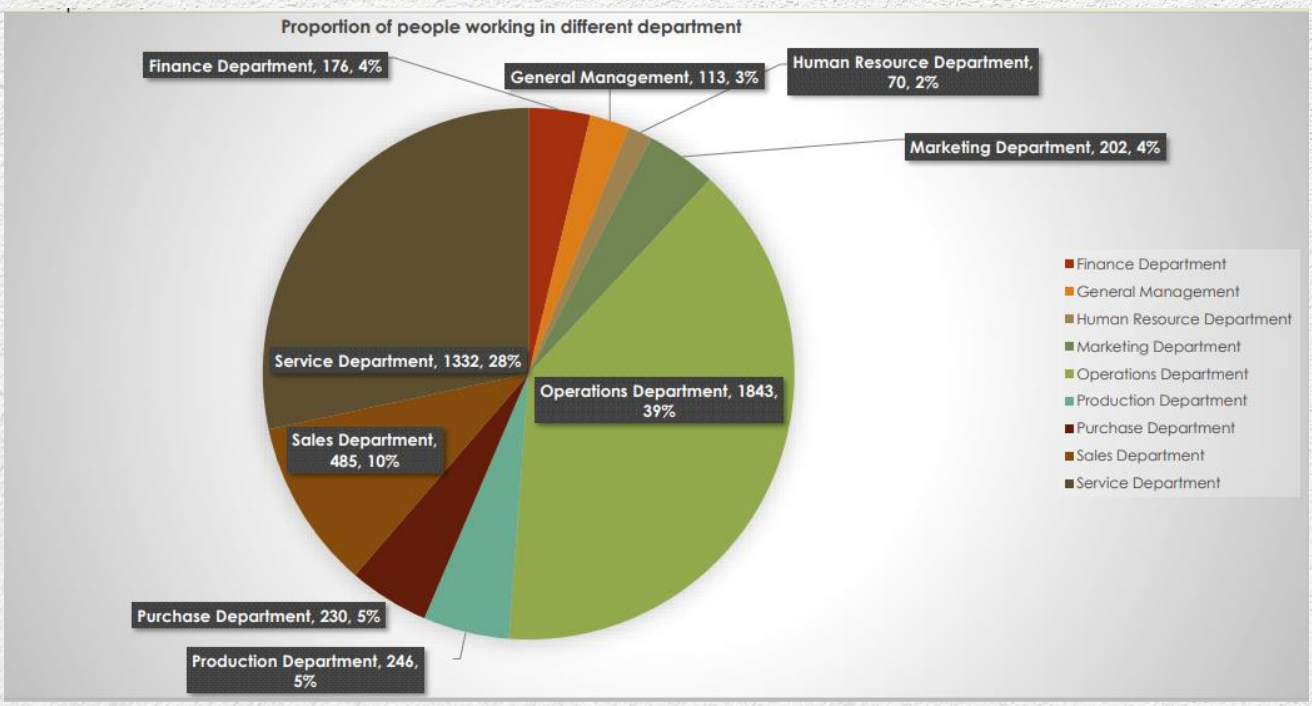
Output/Result : 49983.03223

Q.3 Draw the class intervals for salary in the company ?

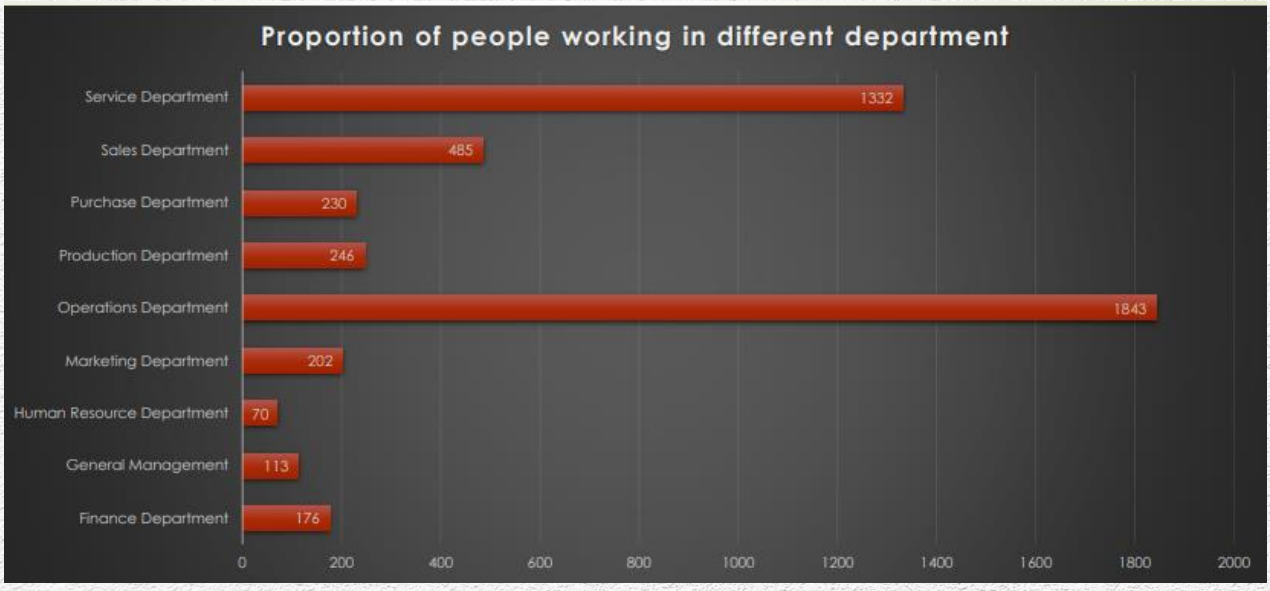


Q.4 Draw Pie Chart / Bar Graph (or any other graph) to show proportion of people working different department ?

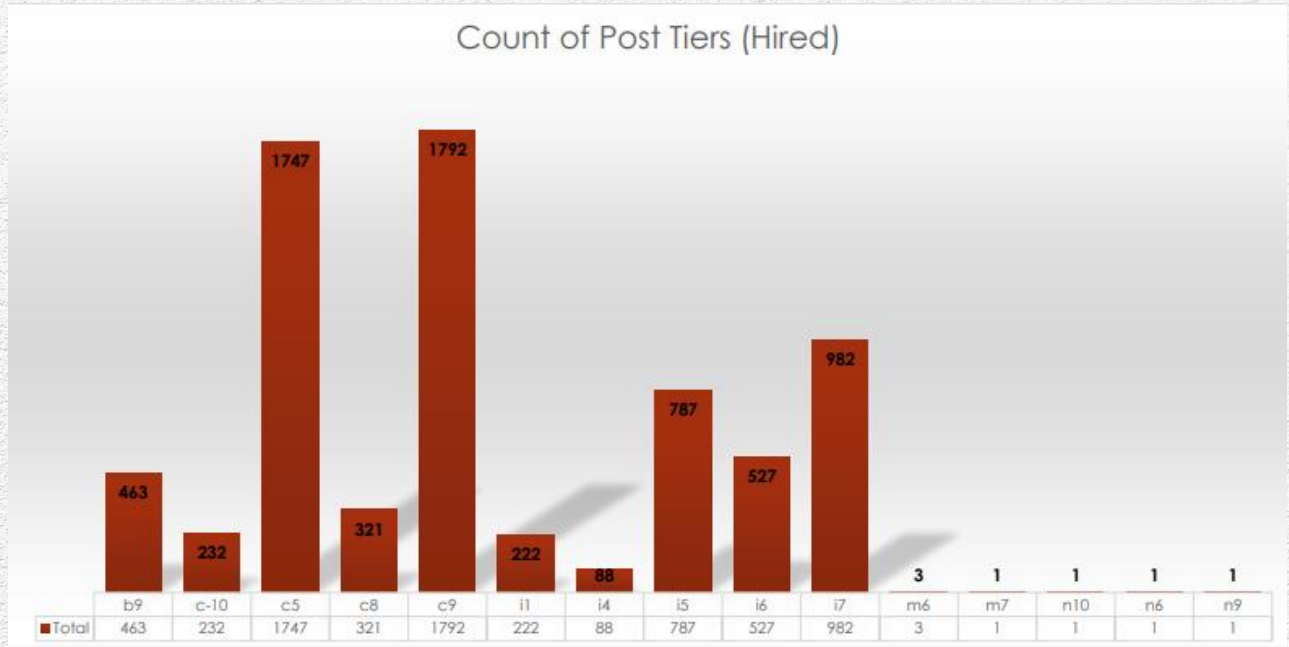
Department	Status	Count of Department
Finance Department		176
General Management		113
Human Resource Department		70
Marketing Department		202
Operations Department		1843
Production Department		246
Purchase Department		230
Sales Department		485
Service Department		1332

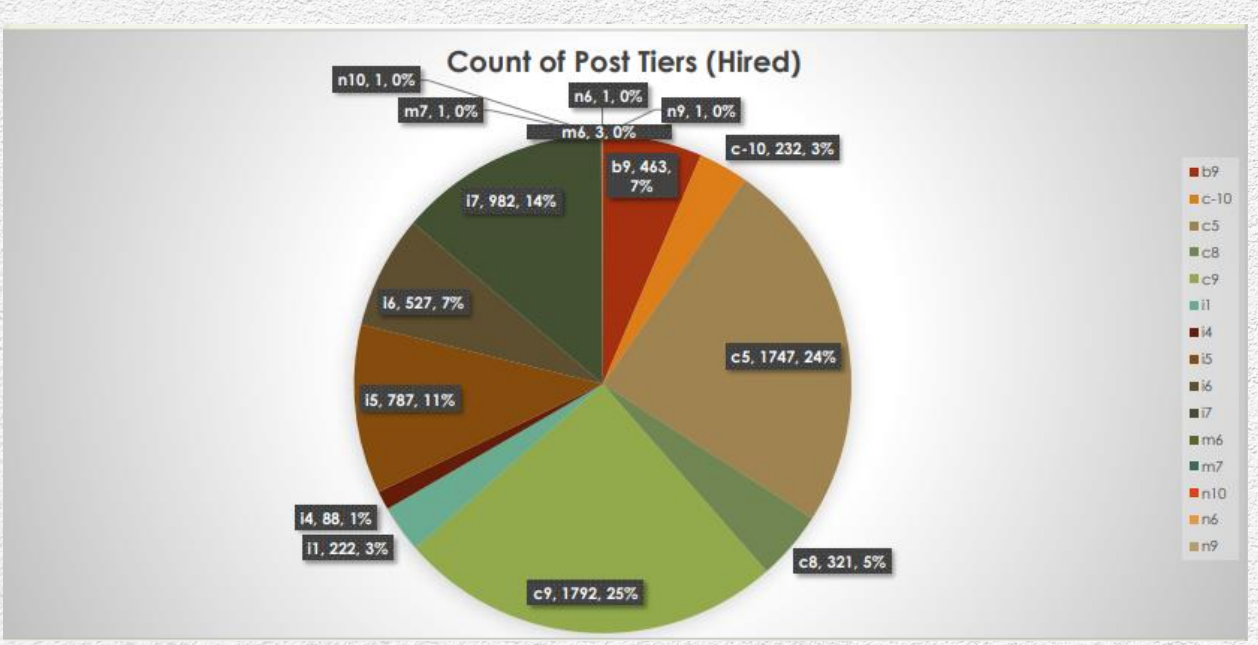


Q.5 Represent different post tiers using chart/graph.

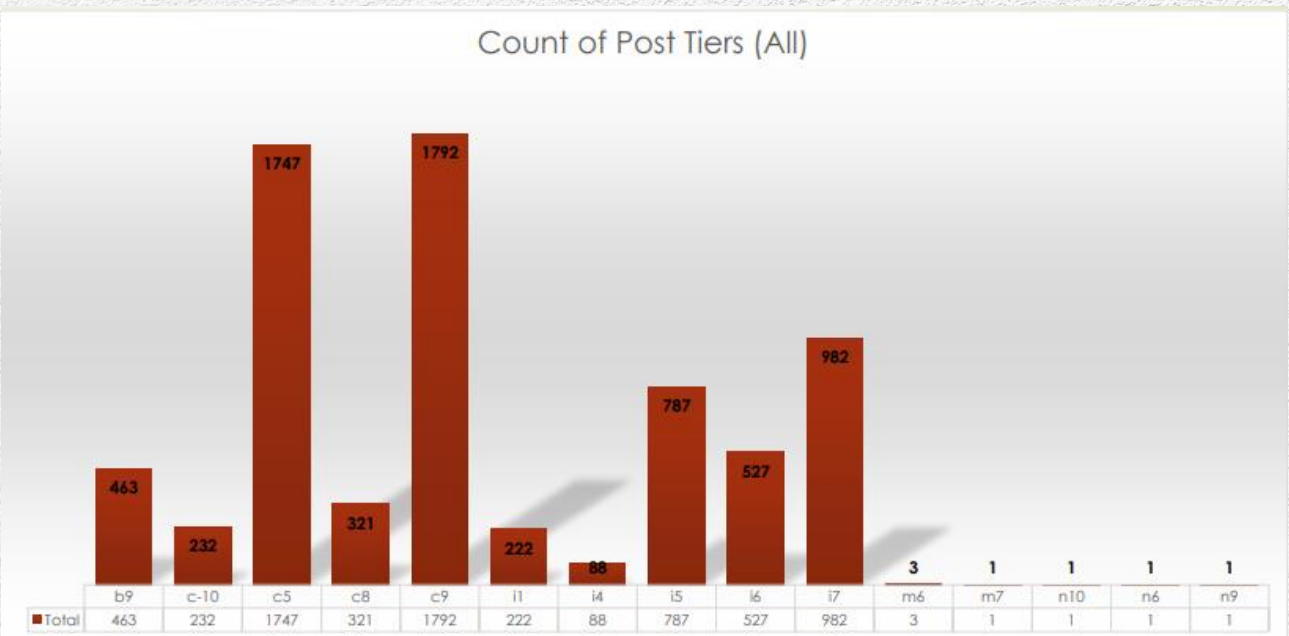


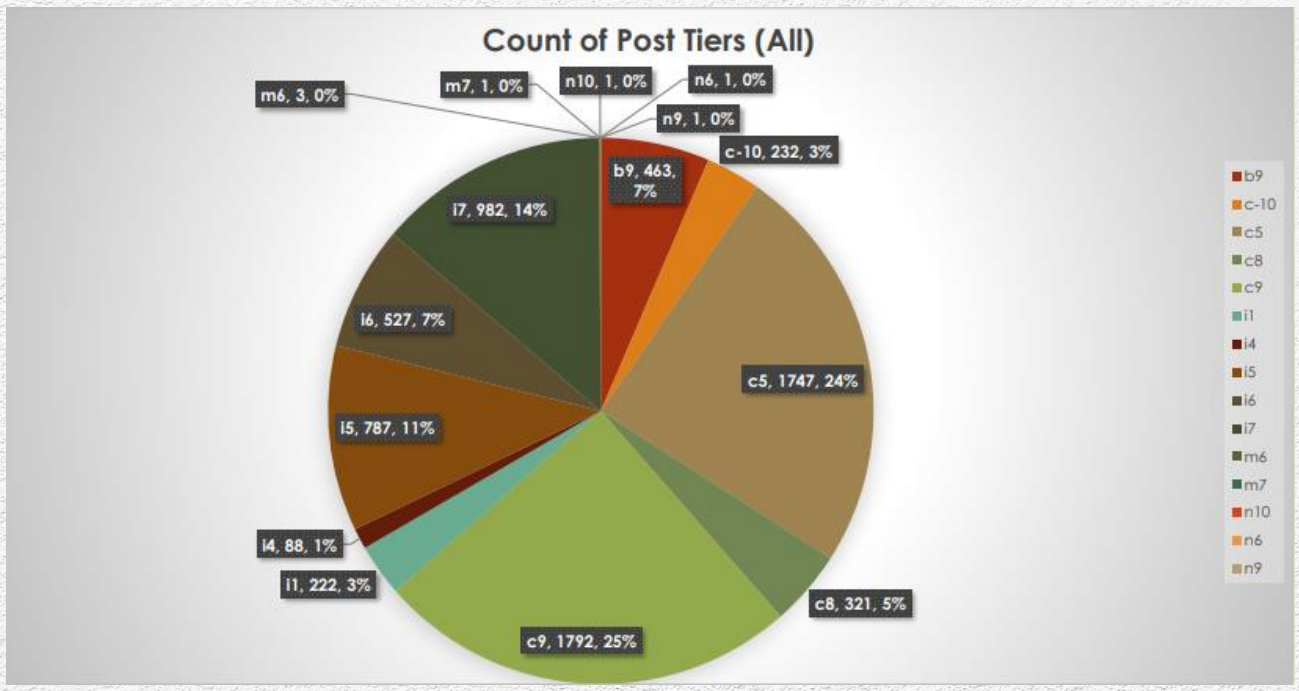
Post Name	Status	Count of Post Tiers (Hired)
b9		308
c-10		105
c5		1182
c8		194
c9		1239
i1		151
i4		32
i5		511
i6		337
i7		635
m6		2
n6		1


Count of Post Tiers (Hired)
463
232
1747
321
1792
222
88
787
527
982
3
1
1
1
1
b9
c-10
c5
c8
c9
i1
i4
i5
i6
i7
m6
m7
n10
n6
n9
Total
463
232
1747
321
1792
222
88
787
527
982
3
1
1
1
1


Count of Post Tiers (Hired)
n10, 1, 0%
n6, 1, 0%
m7, 1, 0%
n9, 1, 0%
m6, 3, 0%
c-10, 232, 3%
b9, 463, 7%
i7, 982, 14%
i6, 527, 7%
c5, 1747, 24%
i5, 787, 11%
i4, 88, 1%
i1, 222, 3%
c8, 321, 5%
c9, 1792, 25%
b9
c-10
c5
c8
c9
i1
i4
i5
i6
i7
m6
m7
n10
n6
n9

Post Name	Status	Count of Post Tiers (All)
b9		463
c-10		232
c5		1747
c8		321
c9		1792
i1		222
i4		88
i5		787
i6		527
i7		982
m6		3
m7		1
n10		1
n6		1
n9		1





Google Drive Link for the Updated and edited Excel sheet- https://docs.google.com/spreadsheets/d/1TGstjS691ghu_J58mdJ_KKg8R_pDgeVf/edit?usp=sharing&ouid=112551872580189129915&rtpof=true&sd=true

IMDB Movie Analysis

Description

"What factors influence the success of a movie on IMDB?"

Here, success can be defined by high IMDB ratings. The impact of this problem is significant for movie producers, directors, and investors who want to understand what makes a movie successful to make informed decisions in their future projects.

Approach

For this project, first, we'll get an understanding of the given data. Then We'll clean the data as per our requirement by removing null values, deleting unnecessary columns, etc. After the cleaning, we'll use a pivot table, various functions, and charts for desired answers to the questions. We'll continue to ask Whys to data to get in-depth of the root of the problem. In the end, we'll present our answers with proper formatting in tables and graphs.

Tech- Stack Used

For this IMBD Movie Analytics project, I used the Office 365 suite's Microsoft Excel. The Office 365 suite is a comprehensive collection of products offered by Microsoft Corporation. It is a productivity-focused suite that assists people and businesses in carrying out and managing a variety of daily tasks and data..

INSIGHTS

1. **Cleaning the data:** This is one of the most important steps to perform before moving forward with the analysis. Use your knowledge learned till now to do this. (Dropping columns, removing null values, etc.)

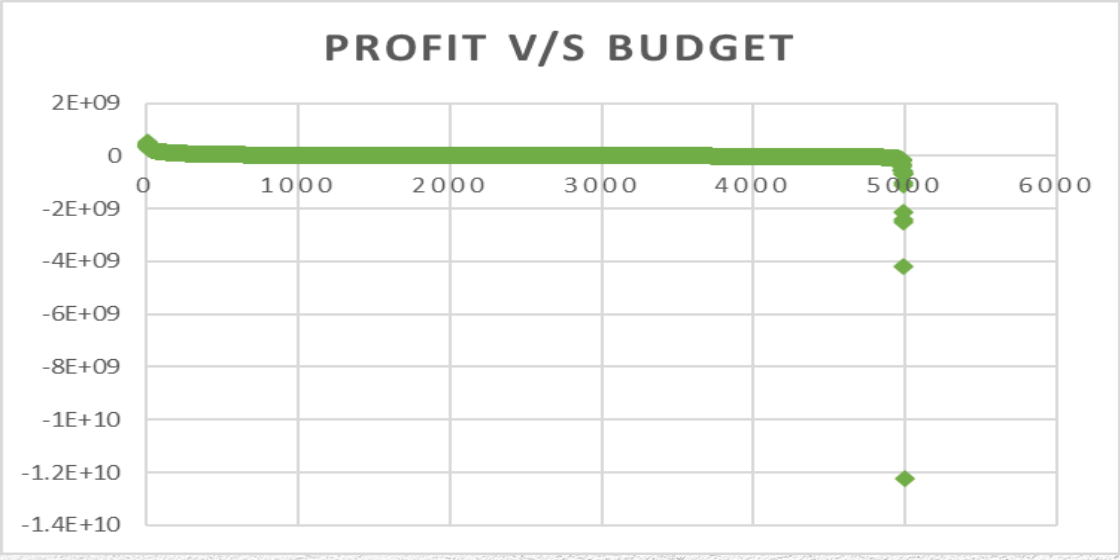
Your task: Clean the data.

* Dropping unnecessary columns.
(Color, director_facebook_likes, actor_3_facebook_likes, actor_2_name, actor_1_facebook_likes, cast_total_facebook_likes, actor_3_name, facenumber_in_posts, plot_keywords, movie_imdb_link, content_rating, actor_2_facebook_likes, aspect_ratio, movie_facebook_likes)

* Remove Blank Cell / Null Value.
* Removing Duplicate.

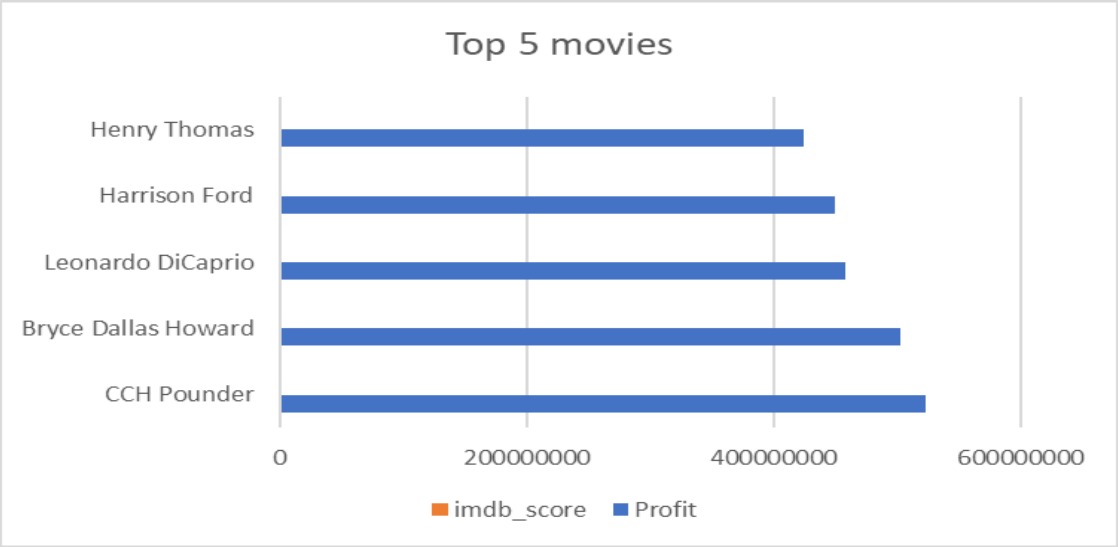
2. **Movies with the highest profit:** Create a new column called profit which contains the difference between the two columns: gross and budget. Sort the column using the profit column as a reference. Plot profit (y-axis) vs budget (x-axis) and observe the outliers using the appropriate chart type.
Your task: Find the movies with the highest profit

Profit Vs Budget



Top 5 Profitable Movies

director_name	genres	actor_1_name	Movie_name	language	Profit	imdb_score
James Cameron	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	AvatarÂ	English	523505847	7.9
Colin Trevorrow	Action|Adventure|Sci-Fi|Thriller	Bryce Dallas Howard	Jurassic WorldÂ	English	502177271	7
James Cameron	Drama|Romance	Leonardo DiCaprio	TitanicÂ	English	458672302	7.7
George Lucas	Action|Adventure|Fantasy|Sci-Fi	Harrison Ford	Star Wars: Episode IV - A New HopeÂ	English	449935665	8.7
Steven Spielberg	Family|Sci-Fi	Henry Thomas	E.T. the Extra-TerrestrialÂ	English	424449459	7.9



C. Top 250 Movies:

Create a new column IMDb_Top_250 and store the top 250 movies with the highest IMDb Rating (corresponding to the column: imdb_score). Also make sure that for all of these movies, the num_voted_users is greater than 25,000. Also add a Rank column containing the values 1 to 250 indicating the ranks of the corresponding films.

Extract all the movies in the IMDb_Top_250 column which are not in the English language and store them in a new column named Top_Foreign_Lang_Film. You can use your own imagination also!

Your task: Find IMDB Top 250

Filter out data where num_voted_users > 25,000 using filter.

Sort the data using the imbd_score column in descending order

Use first 250 entry for our analysis.

We'll give Rank using a Sequence Formula.

=SEQUENCE(COUNTA(G2:G251),1,1,1)

Filter out language by unselecting English. Which gives us foreign language movies in our Top 250 list.

5749

Top 250 Movies:

https://docs.google.com/spreadsheets/d/1jp2CEfhfAZuyCGyuYQ0P65SlZ2aOtb8D/edit?usp=sharing&ouid=113657493328297171808&rtpof=true&sd=true

Top 250 Foreign Language Movies:

https://docs.google.com/spreadsheets/d/1jp2CEfhfAZuyCGyuYQ0P65SlZ2aOtb8D/edit?usp=sharing&ouid=113657493328297171808&rtpof=true&sd=true

D.Top 10 Best Directors:

Group the column using the director_name column.

Find out the top 10 directors for whom the mean of imdb_score is the highest and store them in a new column top10director.
In case of a tie in IMDb score between two directors, sort them alphabetically.

Your task: Find the best directors.

Using Pivot Table, Filter, and Sorting.

Top 10 Directors	Average of imdb_score
Charles Chaplin	8.60
Tony Kaye	8.60
Alfred Hitchcock	8.50
Damien Chazelle	8.50
Majid Majidi	8.50
Ron Fricke	8.50
Sergio Leone	8.43
Christopher Nolan	8.43
Asghar Farhadi	8.40
Marius A. Markevicius	8.40

Top 10 Directors
https://docs.google.com/spreadsheets/d/1jp2CEfhfAZuyCGyuYQ0P65SlZ2aOtb8D/edit?usp=sharing&ouid=113657493328297171808&rtpof=true&sd=true

E. Popular Genres:

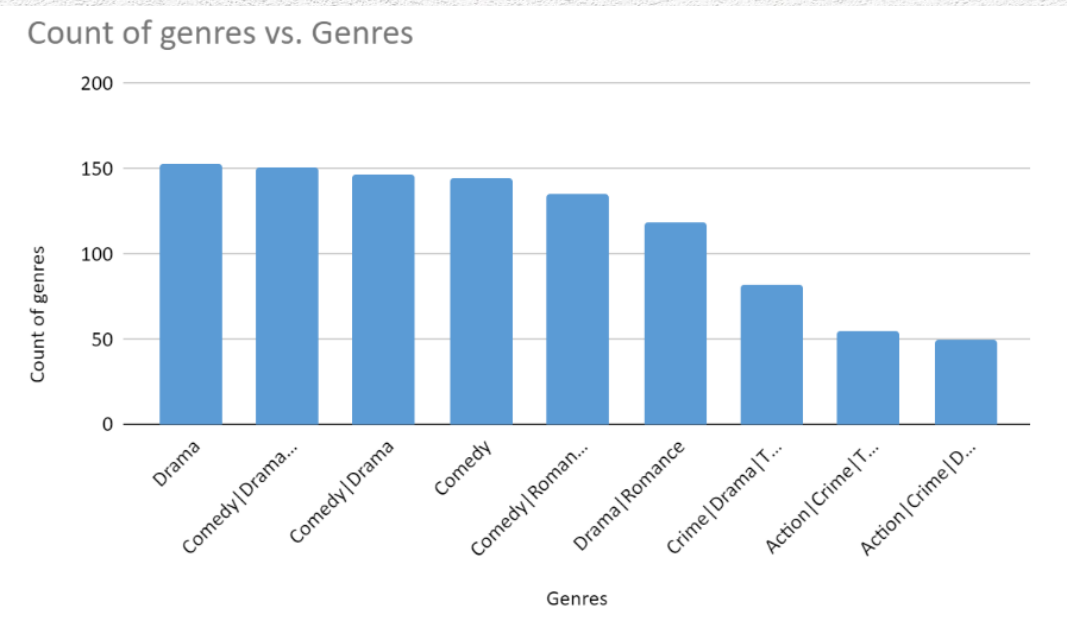
Perform this step using the knowledge gained while performing previous steps.

Your task: Find popular genres.

Using Pivot table, Filter, and Sorting.

Genres	Count of genres
Drama	153
Comedy|Drama|Romance	151
Comedy|Drama	147
Comedy	145
Comedy|Romance	135
Drama|Romance	119
Crime|Drama|Thriller	82
Action|Crime|Thriller	55
Action|Crime|Drama|Thriller	50

Most popular genres



We can see that Drama is most popular genre here.

https://docs.google.com/spreadsheets/d/1jp2CEfhfAZuyCGyuYQ0P65SlZ2aOtb8D/edit?usp=sharing&ouid=113657493328297171808&rtpof=true&sd=true

F. Find the mean of the num_critic_for_reviews and num_users_for_review and identify the actors which have the highest mean.

Charts: Create three new columns namely,

Meryl_Streep, Leo_Caprio,and Brad_Pitt which contain the movies in which the actors: 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' are the lead actors. Use only the actor_1_name column for extraction. Also, make sure that you use the names 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' for the said extraction.

Append the rows of all these columns and store them in a new column named Combined.

Group the combined column using the actor_1_name column.

Find the mean of the num_critic_for_reviews and num_users_for_review and identify the actors which have the highest mean.

Observe the change in number of voted users over decades using a bar chart. Create a column called decade which represents the decade to which every movie belongs to. For example, the title_year year 1923, 1925 should be stored as 1920s. Sort the column based on the column decade, group it by decade and find the sum of users voted in each decade. Store this in a new data frame called df_by_decade.
Your task: Find the critic-favorite and audience-favorite actors.

Using Pivot table.

actor_1_name	Mean of num_user_for_reviews	Mean of num_critic_for_reviews
Brad Pit	742.35	245.00
Leonardo DiCaprio	914.48	330.19
Meryl Streep	297.18	181.45

TOP 3 FAMOUS ACTORS

Mean of num_user_for_reviews Mean of num_critic_for_reviews

Here We can see that Leonardo DiCaprio is the audience's and Critic's favorite actor.

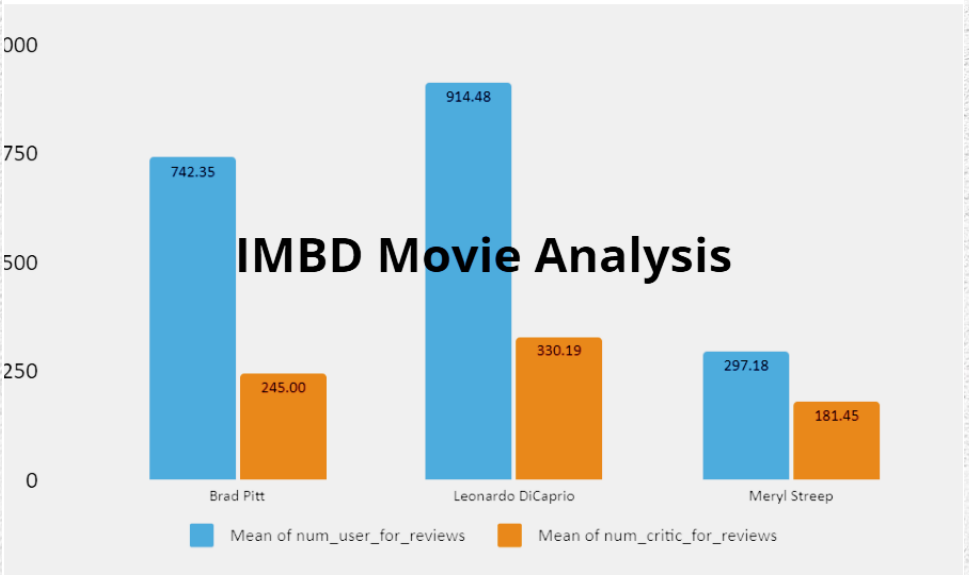
User voting by decade:

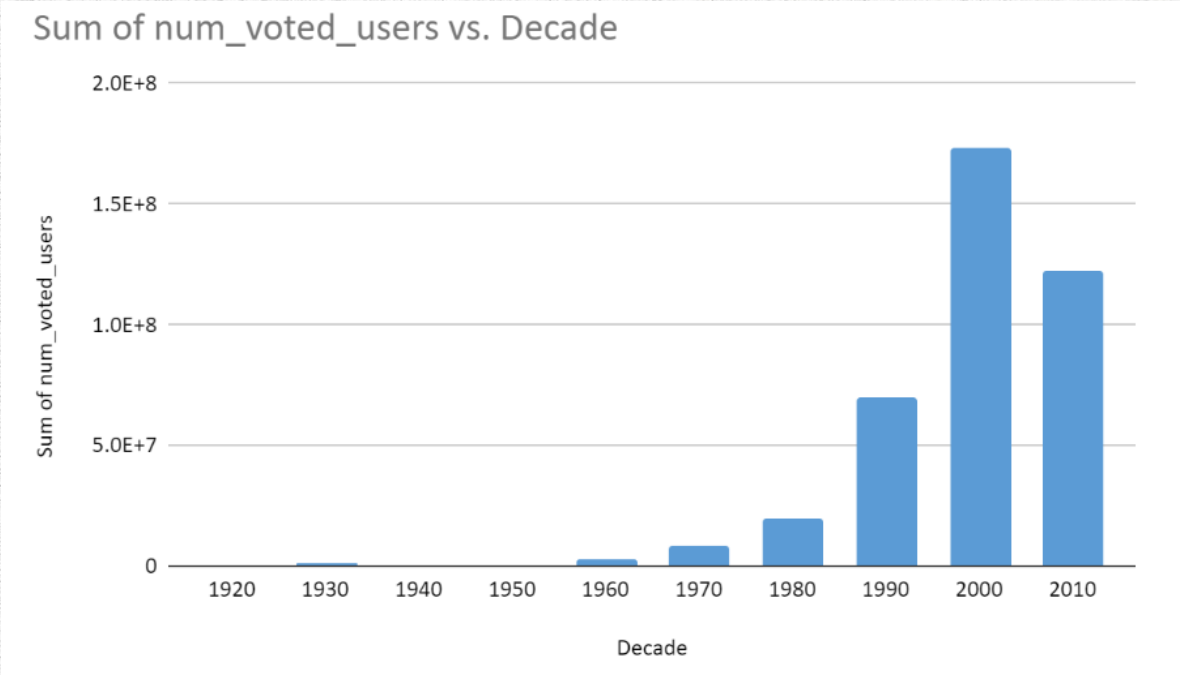
By using a pivot table.

Decade	Sum of num_voted_users
1920	116392
1930	804839
1940	230838
1950	678336
1960	2985581
1970	8704723
1980	20101705
1990	70090204
2000	173033966
2010	122492496

INCREASE OF VOTED USERS

Chart:





https://docs.google.com/spreadsheets/d/1jp2CEfhfAZuyCGyuYQ0P65SlZ2aOtb8D/edit?usp=sharing&ouid=113657493328297171808&rtpof=true&sd=true

Bank Loan Case Study

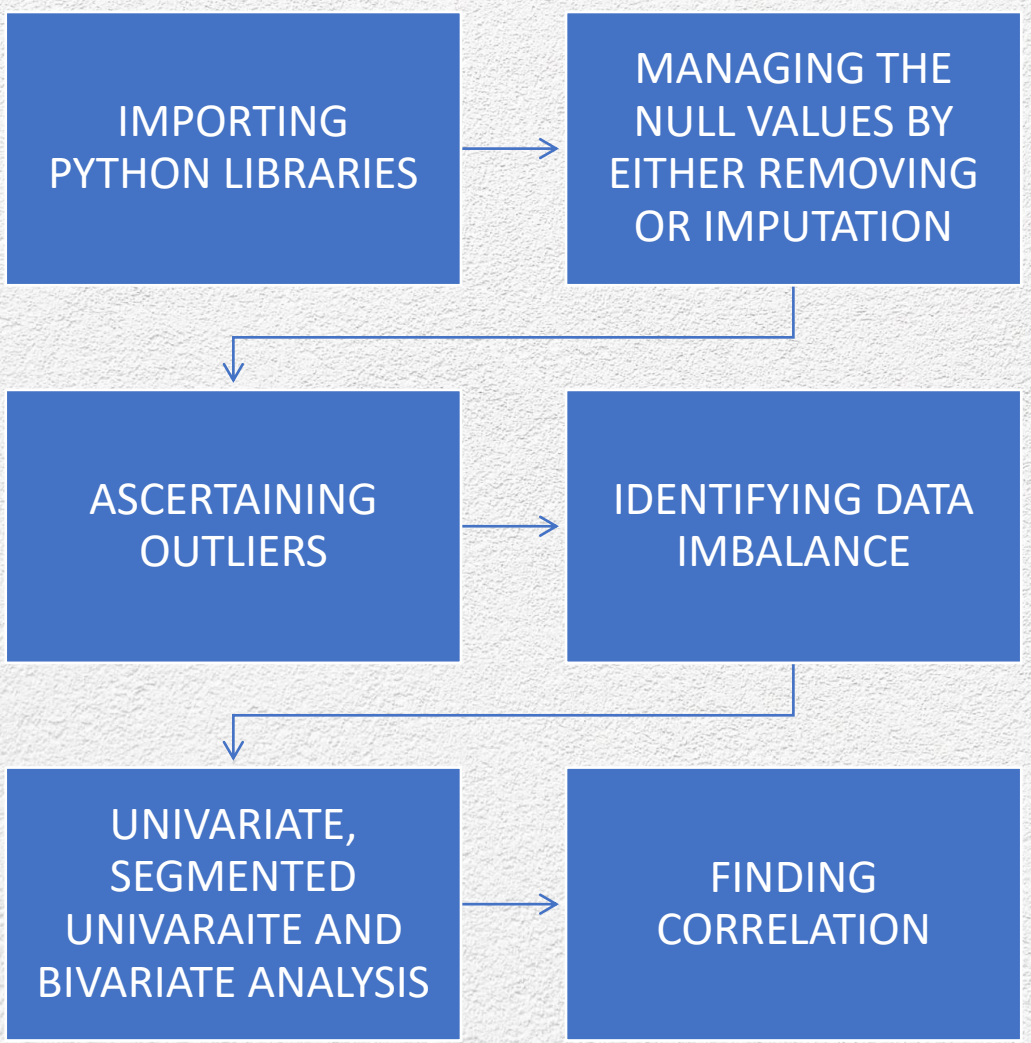
Description

The given project asks us to apply Risk Analytics by analysing the data given for bank loan.
The project expects us to perform Exploratory Data Analysis to understand how consumer attribute which influence the tendency to default.

By EDA we will identify patterns of loan default and the driving factors which leads to these defaults. For this which has strong indicators of default are found.

We are expected to manage missing data, identify outliers, report any data imbalance, explain the result of univariate, segmented univariate, bivariate analysis and determine the top 10 correlation for the client with payment difficulties and all other cases.

APPROACH


IMPORTING PYTHON LIBRARIES
MANAGING THE NULL VALUES BY EITHER REMOVING OR IMPUTATION
ASCERTAINING OUTLIERS
IDENTIFYING DATA IMBALANCE
UNIVARIATE, SEGMENTED UNIVARAITE AND BIVARIATE ANALYSIS
FINDING CORRELATION

TECH STACH USED

JUPYTER Labs was used because they can help in easy python compilations and visualisation.

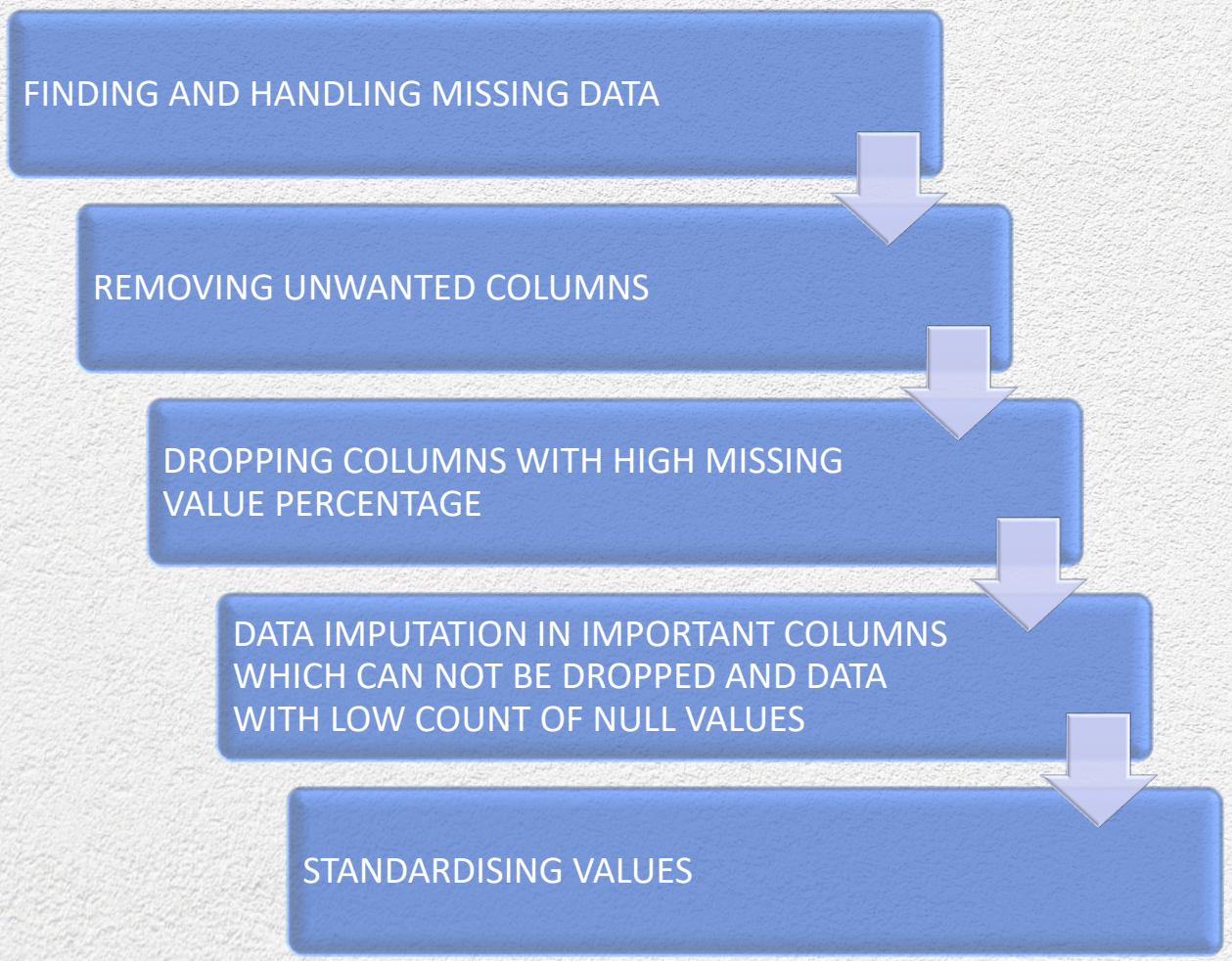
The file can be viewed in the link below:-
Click here to see the IPYNB file

Due to size constraint please don't hesitate to view the visualisations in the IPYNB file.

INSIGHTS

INSIGHTS

TASK 1-DATA CLEANING AND MANIPULATION



FINDING AND HANDLING MISSING DATA

DATASET -1 ->APPLICATION DATA

```
#checking rows and columns
print ("application_data:",apd.shape)
```

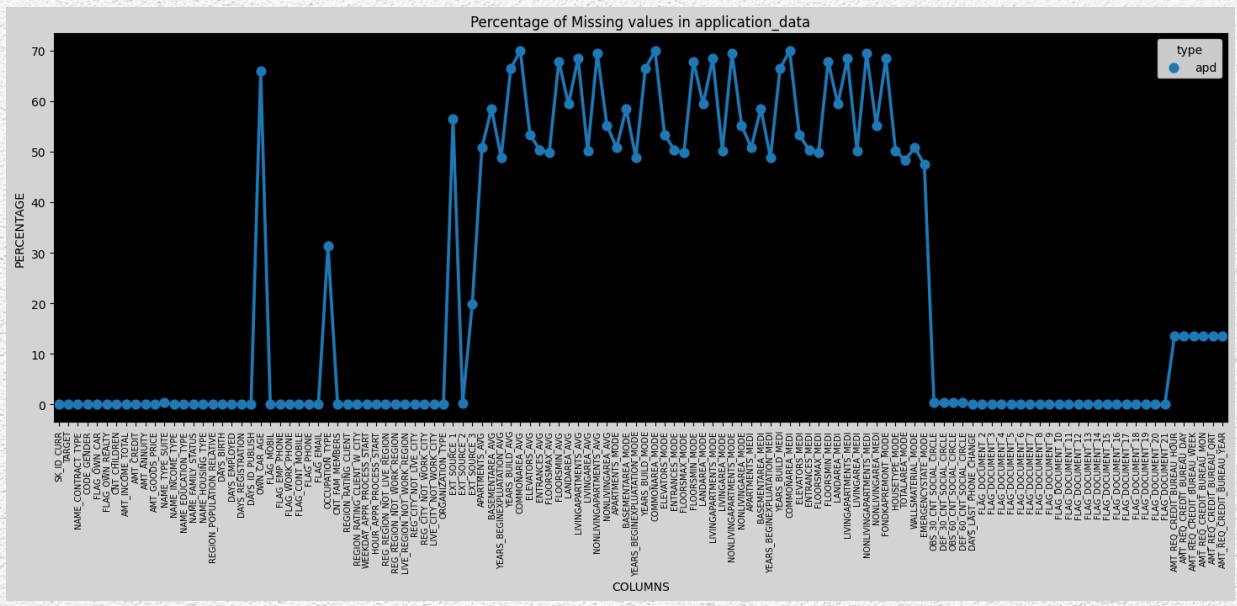
```
application_data: (307511, 122)
```

Number of Rows =307511
Number of Columns=122

There are some data set related to days which is negative would have to manage that

The given line plot shows the columns and number of null values(extracted from the Jupyter File attached in Tech Stack used slide)
We will drop those columns with missing value % greater than 40%

As per Industrial Standard, max Threshold limit can be between 40% to 50 % depending upon the data acquired in specific sector



There are 49 col with null values greater than 40%
After dropping 49 col new shape is 307511 rows and 73 columns

```
apd.shape
```

```
(307511, 73)
```

Remaining columns with null value

```
nullval(apd)[nullval(apd)>0]
```

```
OCCUPATION_TYPE                31.35
EXT_SOURCE_3                   19.83
AMT_REQ_CREDIT_BUREAU_YEAR     13.50
AMT_REQ_CREDIT_BUREAU_QRT      13.50
AMT_REQ_CREDIT_BUREAU_MON      13.50
AMT_REQ_CREDIT_BUREAU_WEEK     13.50
AMT_REQ_CREDIT_BUREAU_DAY      13.50
AMT_REQ_CREDIT_BUREAU_HOUR     13.50
NAME_TYPE_SUITE                 0.42
OBS_30_CNT_SOCIAL_CIRCLE        0.33
DEF_30_CNT_SOCIAL_CIRCLE        0.33
OBS_60_CNT_SOCIAL_CIRCLE        0.33
DEF_60_CNT_SOCIAL_CIRCLE        0.33
EXT_SOURCE_2                    0.21
AMT_GOODS_PRICE                 0.09
dtype: float64
```

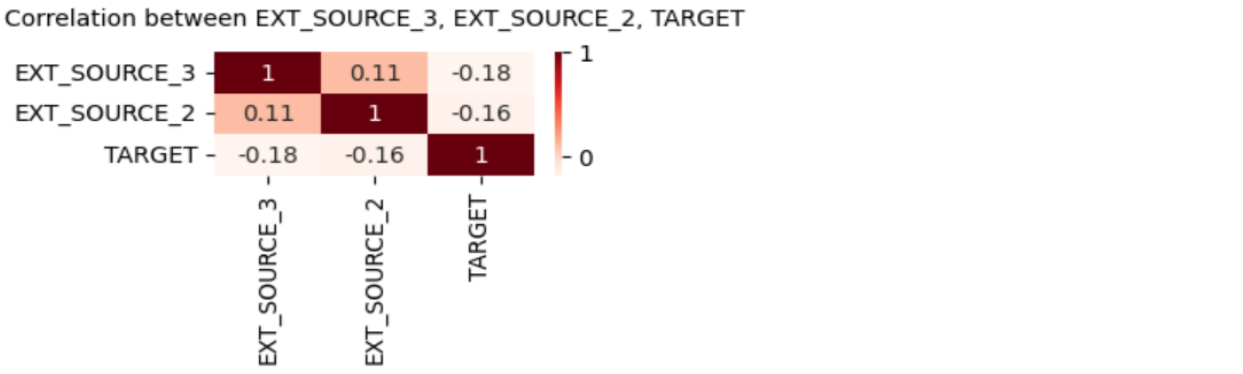
As in the dictionary we see a normalised data set of ext source 1, because ext source 2 and ext source 3 have no linear correlation with target the the column ext source 2 and ext source 3 are dropped
After dropping we have 71 columns

```
(307511, 71)
```

```
#creating a variable null_col_40 for storing null columns having missing values more than 40%
null_col_40 = nullval(apd)[nullval(apd)>40]
print("following columns have null value more than 40%")
print(null_col_40.index)
print("No. of columns with more than 40% missing value:",len(null_col_40.index))
```

```
following columns have null value more than 40%
Index(['COMMONAREA_MEDI', 'COMMONAREA_AVG', 'COMMONAREA_MODE',
       'NONLIVINGAPARTMENTS_MODE', 'NONLIVINGAPARTMENTS_AVG',
       'NONLIVINGAPARTMENTS_MEDI', 'FONDKAPREMONT_MODE',
       'LIVINGAPARTMENTS_MODE', 'LIVINGAPARTMENTS_AVG',
       'LIVINGAPARTMENTS_MEDI', 'FLOORSMIN_AVG', 'FLOORSMIN_MODE',
       'FLOORSMIN_MEDI', 'YEARS_BUILD_MEDI', 'YEARS_BUILD_MODE',
       'YEARS_BUILD_AVG', 'OWN_CAR_AGE', 'LANDAREA_MEDI', 'LANDAREA_MODE',
       'LANDAREA_AVG', 'BASEMENTAREA_MEDI', 'BASEMENTAREA_AVG',
       'BASEMENTAREA_MODE', 'EXT_SOURCE_1', 'NONLIVINGAREA_MODE',
       'NONLIVINGAREA_AVG', 'NONLIVINGAREA_MEDI', 'ELEVATORS_MEDI',
       'ELEVATORS_AVG', 'ELEVATORS_MODE', 'WALLSMATERIAL_MODE',
       'APARTMENTS_MEDI', 'APARTMENTS_AVG', 'APARTMENTS_MODE',
       'ENTRANCES_MEDI', 'ENTRANCES_AVG', 'ENTRANCES_MODE', 'LIVINGAREA_AVG',
       'LIVINGAREA_MODE', 'LIVINGAREA_MEDI', 'HOUSETYPE_MODE',
       'FLOORSMAX_MODE', 'FLOORSMAX_MEDI', 'FLOORSMAX_AVG',
       'YEARS_BEGINEXPLUATATION_MODE', 'YEARS_BEGINEXPLUATATION_MEDI',
       'YEARS_BEGINEXPLUATATION_AVG', 'TOTALAREA_MODE', 'EMERGENCYSTATE_MODE'],
      dtype='object')
No. of columns with more than 40% missing value: 49
```

```
#removing extra columns
#putting irrelevent columns in 1 variable
irrev = ["EXT_SOURCE_3","EXT_SOURCE_2"]
#making a heatmap
plt.figure(figsize= [3,1])
sns.heatmap(apd[irrev+["TARGET"]].corr(), cmap="Reds",annot=True)
plt.title("Correlation between EXT_SOURCE_3, EXT_SOURCE_2, TARGET", fontdict={"fontsize":10}, pad=12)
plt.show()
```



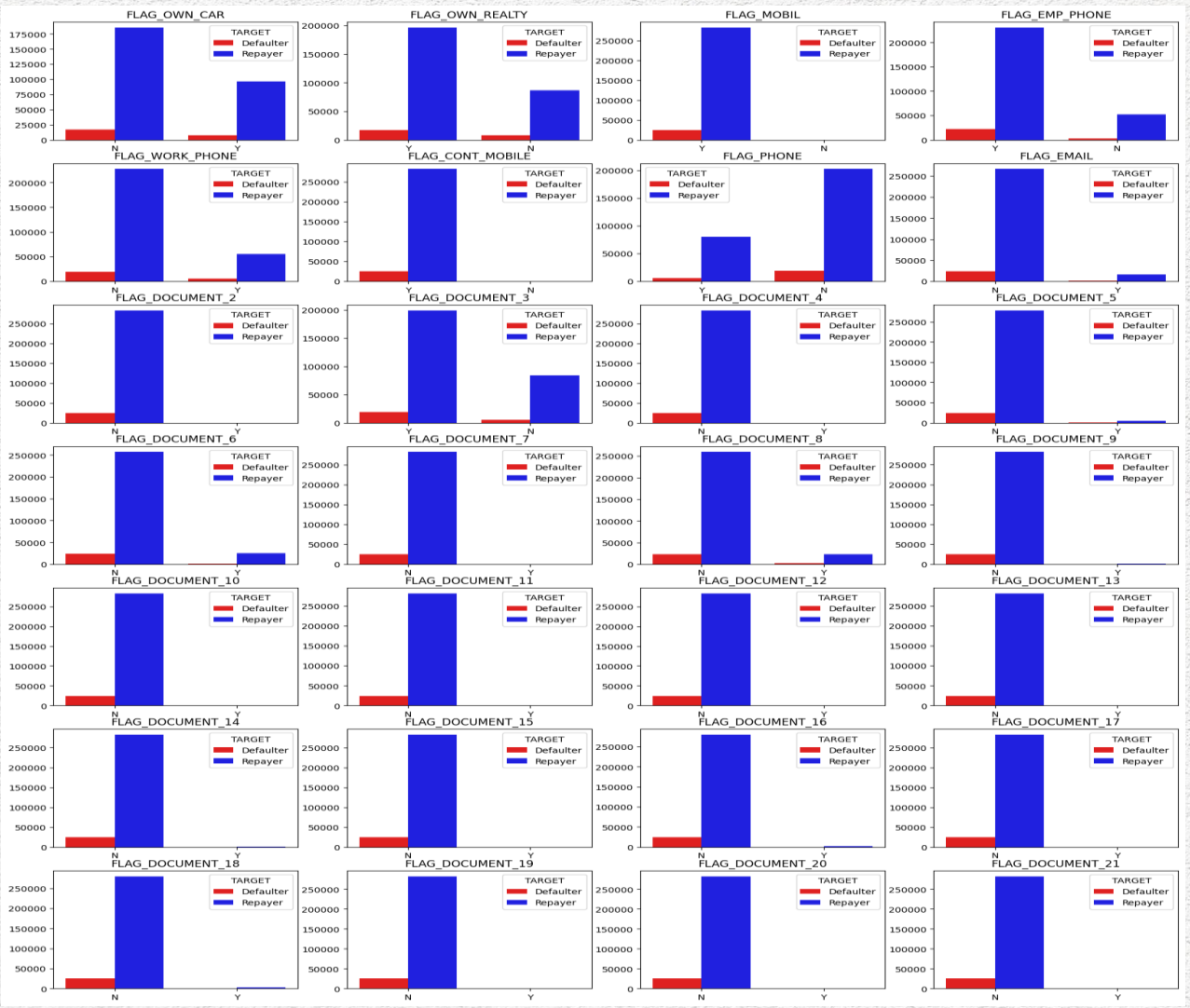
Checking column with FLAGS and determining their relationship with target column
Then checking correlation with target and removing unwanted columns

Columns (FLAG_OWN_REALTY, FLAG_MOBIL ,FLAG_EMP_PHONE, FLAG_CONT_MOBILE, FLAG_DOCUMENT_3) have more **yeses** than **nos** and from these keeping FLAG_DOCUMENT_3,FLAG_OWN_REALTY, FLAG_MOBIL more sense thus we can include these columns and remove all other FLAG columns for further analysis.

Hence we drop these columns and after removing all unnecessary columns now we have 46 relevant columns

```
apd.shape
```

```
(307511, 46)
```


FLAG_OWN_CAR
FLAG_OWN_REALTY
FLAG_MOBIL
FLAG_EMP_PHONE
FLAG_WORK_PHONE
FLAG_CONT_MOBILE
FLAG_PHONE
FLAG_EMAIL
FLAG_DOCUMENT_2
FLAG_DOCUMENT_3
FLAG_DOCUMENT_4
FLAG_DOCUMENT_5
FLAG_DOCUMENT_6
FLAG_DOCUMENT_7
FLAG_DOCUMENT_8
FLAG_DOCUMENT_9
FLAG_DOCUMENT_10
FLAG_DOCUMENT_11
FLAG_DOCUMENT_12
FLAG_DOCUMENT_13
FLAG_DOCUMENT_14
FLAG_DOCUMENT_15
FLAG_DOCUMENT_16
FLAG_DOCUMENT_17
FLAG_DOCUMENT_18
FLAG_DOCUMENT_19
FLAG_DOCUMENT_20
FLAG_DOCUMENT_21
TARGET
Defaulter
Repayer

DATA IMPUTATION-DATASET1 APPLICATION DATA

- In the column occupation type there are 31.35% null values hence we will change those to ‘Unknown’, this unknown column has the highest percentage.

- Similarly in the column Name Type Suite, the missing values will be replaced with “Unaccompanied” which is the mode of the data.

For rest of the columns with null value we will replace them with their median values.

```
apd[amt_credit].median()
```

```
AMT_REQ_CREDIT_BUREAU_YEAR    1.0
AMT_REQ_CREDIT_BUREAU_QRT     0.0
AMT_REQ_CREDIT_BUREAU_MON     0.0
AMT_REQ_CREDIT_BUREAU_WEEK    0.0
AMT_REQ_CREDIT_BUREAU_DAY     0.0
AMT_REQ_CREDIT_BUREAU_HOUR    0.0
dtype: float64
```

```
apd.fillna(apd[amt_credit].median(),inplace=True)
```

```
apd[a].median()
```

```
DEF_60_CNT_SOCIAL_CIRCLE    0.0
OBS_60_CNT_SOCIAL_CIRCLE    0.0
DEF_30_CNT_SOCIAL_CIRCLE    0.0
OBS_30_CNT_SOCIAL_CIRCLE    0.0
dtype: float64
```

```
nullval(apd).head()
```

```
AMT_GOODS_PRICE            0.09
AMT_ANNUITY                0.00
CNT_FAM_MEMBERS            0.00
DAYS_LAST_PHONE_CHANGE     0.00
SK_ID_CURR                 0.00
dtype: float64
```

```
apd["AMT_GOODS_PRICE"].describe()
```

```
count     3.072330e+05
mean      5.383962e+05
std       3.694465e+05
min       4.050000e+04
25%       2.385000e+05
50%       4.500000e+05
75%       6.795000e+05
max       4.050000e+06
Name: AMT_GOODS_PRICE, dtype: float64
```

```
apd["AMT_GOODS_PRICE"].isnull().sum()
```

```
278
```

```
apd.fillna(apd["AMT_GOODS_PRICE"].median(),inplace=True)
```

```
apd["OCCUPATION_TYPE"]=apd["OCCUPATION_TYPE"].fillna("Unknown")
```

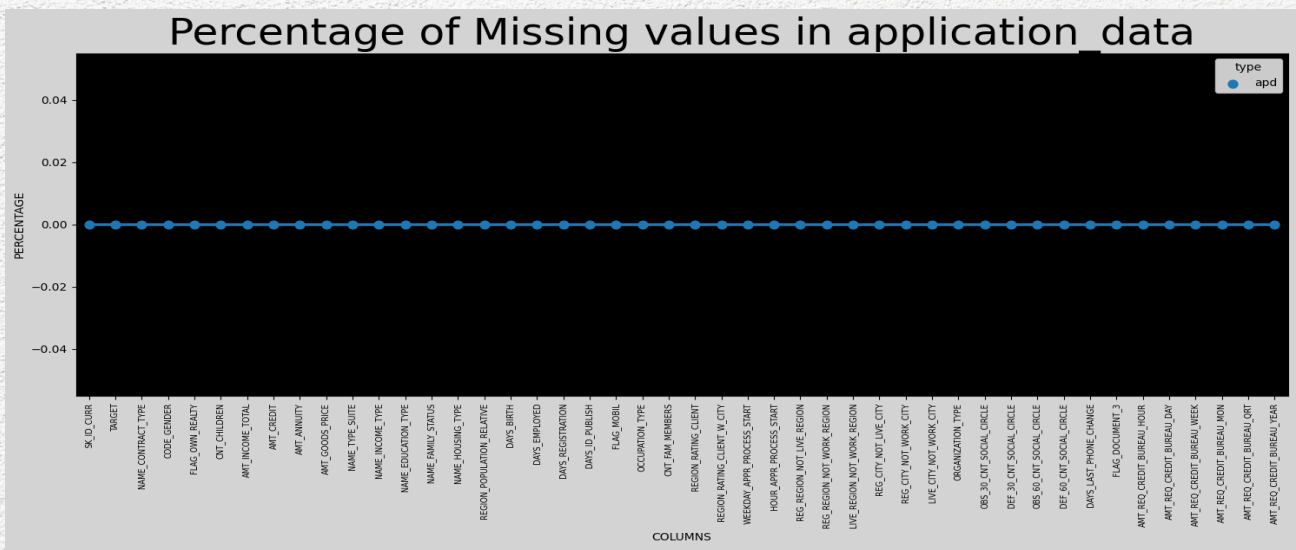
```
apd["OCCUPATION_TYPE"].isnull().sum() #zero null value left
```

```
0
```

```
apd["NAME_TYPE_SUITE"].value_counts()
```

```
Unaccompanied      248526
Family              40149
Spouse, partner     11370
Children             3267
Other_B              1770
Other_A               866
Group of people       271
Name: NAME_TYPE_SUITE, dtype: int64
```

```
apd["NAME_TYPE_SUITE"]=apd["NAME_TYPE_SUITE"].fillna("Unaccompanied")
```



WE CAN SEE NO COLUMN WITH NULL VALUES

STANDARDIZING VALUES – DATASET-1 APPLICATION DATA

```
# Creating bins for Credit amount in term of Lakhs
apd['AMT_CREDIT']=apd['AMT_CREDIT']/100000

bins = [0,1,2,3,4,5,6,7,8,9,10,41] #40.5 is the highest amount
slots = ['0-1L','1L-2L', '2L-3L','3L-4L','4L-5L','5L-6L','6L-7L','7L-8L','8L-9L','9L-10L','10L Above']

apd['AMT_CREDIT_RANGE']=pd.cut(apd['AMT_CREDIT'],bins=bins,labels=slots)
```

```
apd['AMT_CREDIT_RANGE'].value_counts(normalize=True)*100
```

```
2L-3L        17.824728
10L Above    16.254703
5L-6L        11.131960
4L-5L        10.418489
1L-2L         9.801275
3L-4L         8.564897
6L-7L         7.820533
8L-9L         7.086576
7L-8L         6.241403
9L-10L        2.902986
0-1L          1.952450
Name: AMT_CREDIT_RANGE, dtype: float64
```

```
#STANDARDIZING VALUES
# Binning Numerical Columns to create a categorical column
# Creating bins for income amount in term of Lakhs
apd['AMT_INCOME_TOTAL']=apd['AMT_INCOME_TOTAL']/100000
```

```
bins = [0,1,2,3,4,5,6,7,8,9,10,1170]#1170 is the largest value
slot = ['0-1L','1L-2L', '2L-3L','3L-4L','4L-5L','5L-6L','6L-7L','7L-8L','8L-9L','9L-10L','10L Above']

apd['AMT_INCOME_RANGE']=pd.cut(apd['AMT_INCOME_TOTAL'],bins,labels=slot)
```

```
apd["AMT_INCOME_RANGE"].value_counts(normalize=True)*100
```

```
1L-2L        50.696723
2L-3L        21.194689
0-1L         20.714056
3L-4L         4.772512
4L-5L         1.743352
5L-6L         0.356085
6L-7L         0.282592
8L-9L         0.096907
10L Above     0.081298
7L-8L         0.052681
9L-10L        0.009105
Name: AMT_INCOME_RANGE, dtype: float64
```

```
# Creating bins for Price of Goods in term of Lakhs
apd['AMT_GOODS_PRICE']=apd['AMT_GOODS_PRICE']/100000

bins = [0,1,2,3,4,5,6,7,8,9,10,41] #40.5 is the highest value
slots = ['0-1L','1L-2L', '2L-3L','3L-4L','4L-5L','5L-6L','6L-7L','7L-8L','8L-9L','9L-10L','10L Above']

apd['AMT_GOODS_PRICE_RANGE']=pd.cut(apd['AMT_GOODS_PRICE'],bins=bins,labels=slots)
```

```
apd['AMT_GOODS_PRICE_RANGE'].value_counts(normalize=True)*100
```

```
2L-3L         20.409351
4L-5L         18.617545
6L-7L         13.015469
10L Above     11.095213
1L-2L         10.717015
8L-9L          6.986417
3L-4L          6.900241
5L-6L          4.265538
0-1L           2.832094
7L-8L          2.637304
9L-10L         2.523812
Name: AMT_GOODS_PRICE_RANGE, dtype: float64
```

columns DAYS_BIRTH, DAYS_EMPLOYED, DAYS_REGISTRATION, DAYS_ID_PUBLISH, DAYS_LAST_PHONE_CHANGE which counts days have negative values. thus will correct those values

	DAYS_BIRTH	DAYS_EMPLOYED	DAYS_REGISTRATION	DAYS_ID_PUBLISH	DAYS_LAST_PHONE_CHANGE
count	307511.000000	307511.000000	307511.000000	307511.000000	307511.000000
mean	-16036.995067	63815.045904	-4986.120328	-2994.202373	-961.392295
std	4363.988632	141275.766519	3522.886321	1509.450419	1159.717257
min	-25229.000000	-17912.000000	-24672.000000	-7197.000000	-4292.000000
25%	-19682.000000	-2760.000000	-7479.500000	-4299.000000	-1570.000000
50%	-15750.000000	-1213.000000	-4504.000000	-3254.000000	-757.000000
75%	-12413.000000	-289.000000	-2010.000000	-1720.000000	-274.000000
max	-7489.000000	365243.000000	0.000000	0.000000	450000.000000

```
apd[days_col]=abs(apd[days_col])
apd[days_col].describe()
```

	DAYS_BIRTH	DAYS_EMPLOYED	DAYS_REGISTRATION	DAYS_ID_PUBLISH	DAYS_LAST_PHONE_CHANGE
count	307511.000000	307511.000000	307511.000000	307511.000000	307511.000000
mean	16036.995067	67724.742149	4986.120328	2994.202373	964.319019
std	4363.988632	139443.751806	3522.886321	1509.450419	1157.284784
min	7489.000000	0.000000	0.000000	0.000000	0.000000
25%	12413.000000	933.000000	2010.000000	1720.000000	274.000000
50%	15750.000000	2219.000000	4504.000000	3254.000000	757.000000
75%	19682.000000	5707.000000	7479.500000	4299.000000	1570.000000
max	25229.000000	365243.000000	24672.000000	7197.000000	450000.000000

convert DAYS_BIRTH to AGE in years , DAYS_EMPLOYED to YEARS EMPLOYED

```
#days brith to age
apd["AGE"] = abs(apd["DAYS_BIRTH"]/365)
bins = [0,20,25,30,35,40,45,50,55,60,100]
slots = ["0-20","20-25","25-30","30-35","35-40","40-45","45-50","50-55","55-60","60 Above"]

apd["AGE_GROUP"] = pd.cut(apd["AGE"], bins=bins, labels=slots)
```

```
apd["AGE_GROUP"].value_counts(normalize=True)*100
```

```
35-40       13.940314
40-45       13.464884
30-35       12.825557
60 Above    11.569993
45-50       11.425608
50-55       11.362846
55-60       10.770346
25-30       10.686447
20-25        3.954005
0-20         0.000000
Name: AGE_GROUP, dtype: float64
```

```
#creating column "EMPLOYEMENT_YEARS" from "DAYS_EMPLOYED"

apd["YEARS_EMPLOYED"] = apd["DAYS_EMPLOYED"]/365
bins = [0,5,10,15,20,25,30,1001] #1000.5 was largest value
slots = ["0-5","5-10","10-15","15-20","20-25","25-30","30 Above"]

apd["EMPLOYEMENT_YEARS"] = pd.cut(apd["YEARS_EMPLOYED"], bins=bins, labels=slots)
```

```
apd["EMPLOYEMENT_YEARS"].value_counts(normalize=True)*100
```

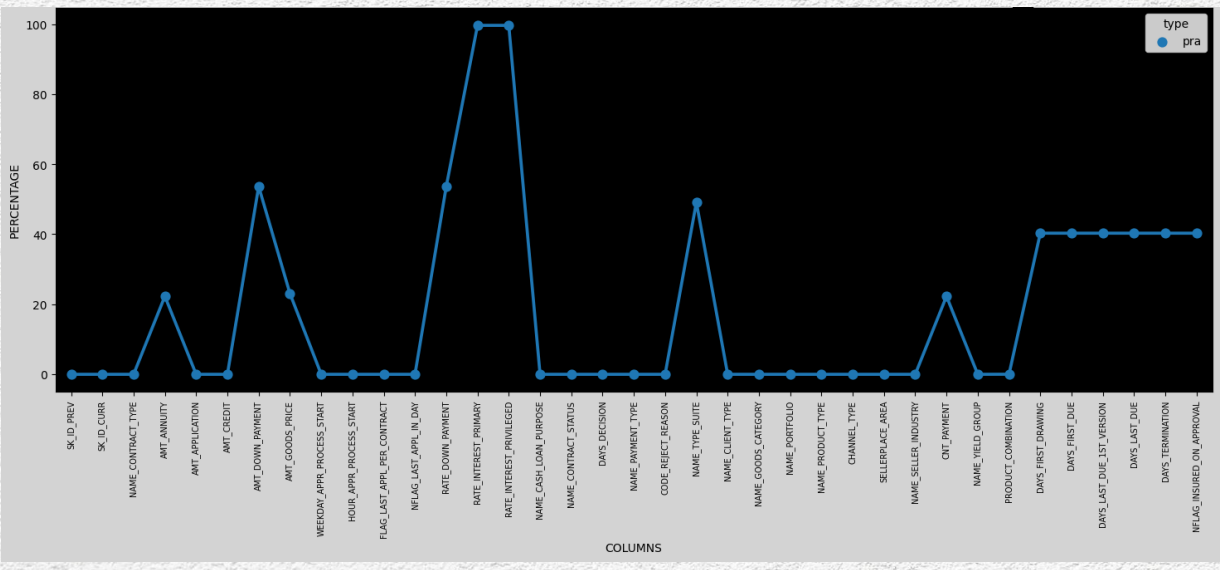
```
0-5         44.326833
5-10        21.095968
30 Above    18.984485
10-15        8.958762
15-20        3.528027
20-25        2.030184
25-30        1.075741
Name: EMPLOYEMENT_YEARS, dtype: float64
```

FINDING AND HANDLING MISSING DATA DATASET -2 ->PREVIOUS APPLICATION

Number of Rows =1670214
Number of Columns=37
There are some data set related to days which is negative would have to manage that
The given line plot shows the columns and number of null values(extracted from the Jupyter File attached in Tech Stack used slide)
We will drop those columns with missing value % greater than 50%
As per Industrial Standard, max Threshold limit can be between 40% to 50 % depending upon the data acquired in specific sector

```
print("Previous Application:", pra.shape)
```

```
Previous Application: (1670214, 37)
```



DROPPING UNWANTED COLUMNS

Removed 4 columns with null values greater than 50% , now there are 33 columns left
Dropped 4 more columns which were not necessary for the analysis, 29 columns
Left

```
# Listing down columns which are not needed

Unnecessary_col = ['WEEKDAY_APPR_PROCESS_START','HOUR_APPR_PROCESS_START','FLAG_LAST_APPL_PER_CONTRACT',
                   'NFLAG_LAST_APPL_IN_DAY']
#pra.drop(Unnecessary_col,axis =1, inplace = True)

pra.shape
```

```
(1670214, 29)
```

```
pnullcol50=nullval(pra)[nullval(pra)>50]
```

```
pnullcol50
```

```
RATE_INTEREST_PRIVILEGED    99.64
RATE_INTEREST_PRIMARY       99.64
AMT_DOWN_PAYMENT            53.64
RATE_DOWN_PAYMENT           53.64
dtype: float64
```

```
pra.drop(columns=pnullcol50.index,inplace=True)
```

```
pra.shape
```

```
(1670214, 33)
```

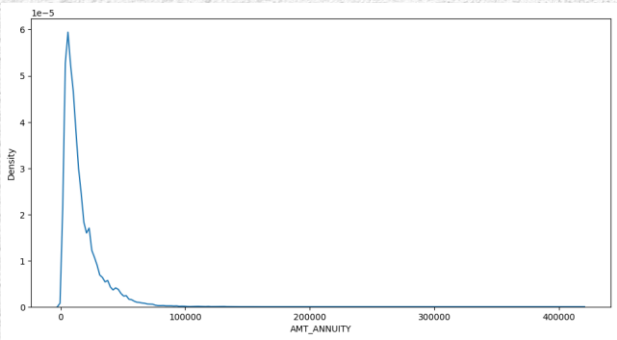
DATA IMPUTATION

In column Name_type_suit, unknown values filled.

In column N_FLAG_INSURED_ON_APPROVAL null values were filled with mode i.e. 0

For AMT_ANNUITY as the kde plot is skewed with 1peak, null values were replaced by median

```
#imputing missing values with median as there is only 1 peak implying outliers

pra['AMT_ANNUITY'].fillna(pra['AMT_ANNUITY'].median(),inplace = True)
```



For AMT_GOOD_PRICE since kdeplot has Multiple peaks, the plot with replacing null values with mode median and mean was made to check which plot resembles the original one the most as it was the mode one null values were replaced with mode

```
# Imputing null values with mode

pra['AMT_GOODS_PRICE'].fillna(pra['AMT_GOODS_PRICE'].mode()[0], inplace=True)
```

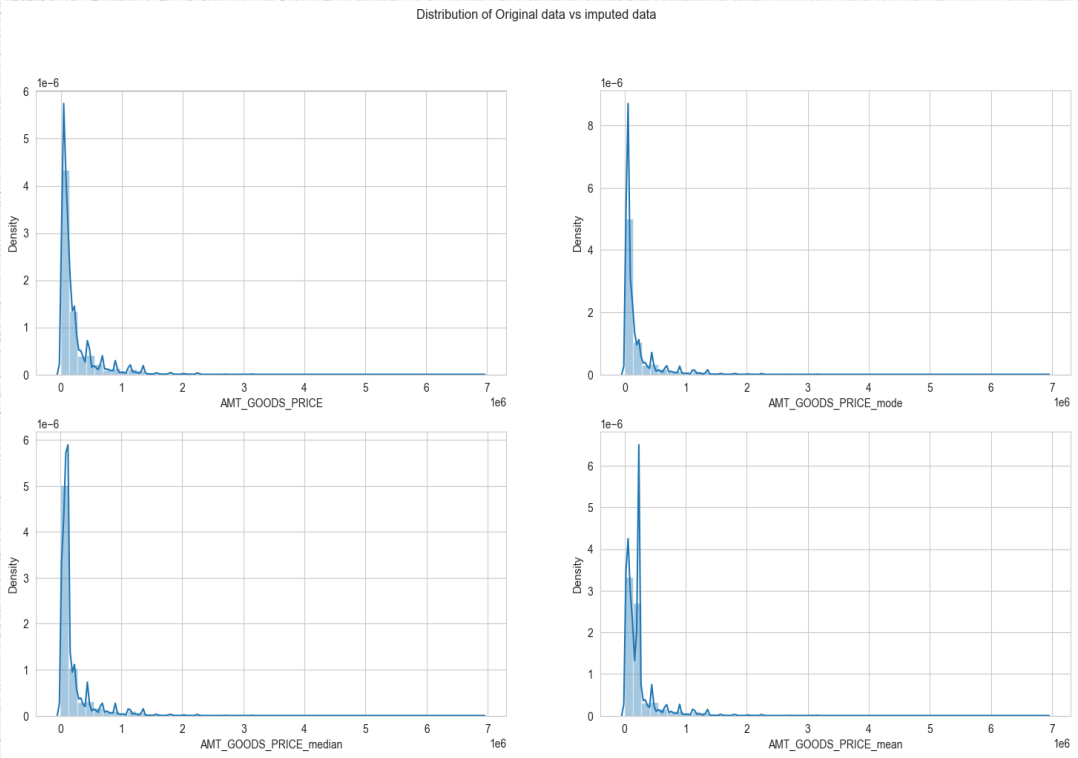
```
# IMputing values "Unknown" as this a categorical column
pra["NAME_TYPE_SUITE"] = pra["NAME_TYPE_SUITE"].fillna("Unknown")

pra["NAME_TYPE_SUITE"].value_counts(normalize=True)*100
```

```
Unknown            49.119754
Unaccompanied      30.473341
Family             12.768603
Spouse, partner     4.015593
Children            1.889937
Other_B             1.055194
Other_A             0.543463
Group of people     0.134115
Name: NAME_TYPE_SUITE, dtype: float64
```

```
pra['NFLAG_INSURED_ON_APPROVAL'].value_counts()
```

```
0.0    665527
1.0    331622
Name: NFLAG_INSURED_ON_APPROVAL, dtype: int64
```

```
pra['NFLAG_INSURED_ON_APPROVAL']=pra['NFLAG_INSURED_ON_APPROVAL'].fillna(0)
```



DATA IMPUTATION
DATASET -2 ->PREVIOUS APPLICATION

Imputing CNT_PAYMENT with 0 as the NAME_CONTRACT_STATUS for these indicate that most of these loans were not started

```
#taking out values count for NAME_CONTRACT_STATUS categories where CNT_PAYMENT have null values.
pra.loc[pra['CNT_PAYMENT'].isnull(),'NAME_CONTRACT_STATUS'].value_counts()
```

```
Canceled        305805
Refused          40897
Unused offer     25524
Approved             4
Name: NAME_CONTRACT_STATUS, dtype: int64
```

```
#imputing null values as 0
pra['CNT_PAYMENT'].fillna(0,inplace = True)
```

In column PRODUCT_COMBINATION the null values were filled with “unknown”

```
pra['PRODUCT_COMBINATION']=pra['PRODUCT_COMBINATION'].fillna("Unkown")
pra['PRODUCT_COMBINATION'].value_counts(normalize=True)*100
```

```
Cash                              17.122956
POS household with interest       15.783726
POS mobile with interest          13.212079
Cash X-Sell: middle                8.614645
Cash X-Sell: low                   7.798282
Card Street                        6.740573
POS industry with interest         5.917385
POS household without interest     4.963915
Card X-Sell                        4.824651
Cash Street: high                  3.570740
Cash X-Sell: high                  3.550503
Cash Street: middle                2.075063
Cash Street: low                   2.025728
POS mobile without interest        1.441851
POS other with interest            1.429697
POS industry without interest      0.754514
POS others without interest        0.152974
Unkown                             0.020716
Name: PRODUCT_COMBINATION, dtype: float64
```

STANDARDIZING VALUES – DATASET-2 PREVIOUS APPLICATION

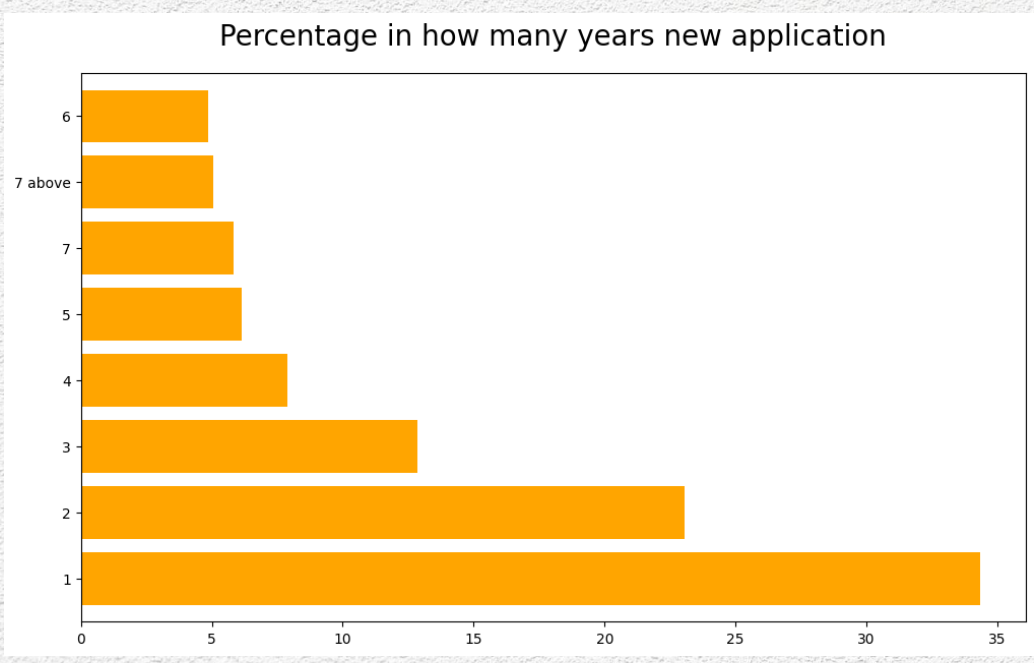
There are negative values in days column which needs to be rectified

	DAYS_DECISION	DAYS_FIRST_DRAWING	DAYS_FIRST_DUE	DAYS_LAST_DUE_1ST_VERSION	DAYS_LAST_DUE	DAYS_TERMI
count	1.670214e+06	997149.000000	997149.000000	997149.000000	997149.000000	99714
mean	-8.806797e+02	342209.855039	13826.269337	33767.774054	76582.403064	8199
std	7.790997e+02	88916.115833	72444.869708	106857.034789	149647.415123	15330
min	-2.922000e+03	-2922.000000	-2892.000000	-2801.000000	-2889.000000	-287
25%	-1.300000e+03	365243.000000	-1628.000000	-1242.000000	-1314.000000	-127
50%	-5.810000e+02	365243.000000	-831.000000	-361.000000	-537.000000	-49
75%	-2.800000e+02	365243.000000	-411.000000	129.000000	-74.000000	-4
max	-1.000000e+00	365243.000000	365243.000000	365243.000000	365243.000000	36524

```
# Converting Negative days to positive days

pra[p_days_col] = abs(pra[p_days_col])

pra[p_days_col].describe() # analysing after conversion
```

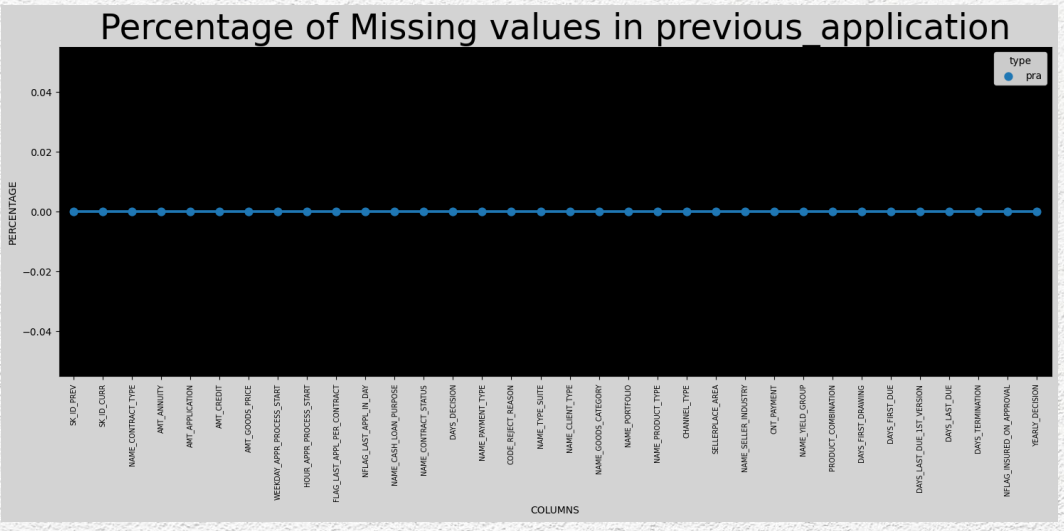
	DAYS_DECISION	DAYS_FIRST_DRAWING	DAYS_FIRST_DUE	DAYS_LAST_DUE_1ST_VERSION	DAYS_LAST_DUE
count	1.670214e+06	997149.000000	997149.000000	997149.000000	997149.000000
mean	8.806797e+02	342340.056543	15949.224065	35163.363265	78152.730207
std	7.790997e+02	88413.495220	72007.270877	106405.950190	148833.342466
min	1.000000e+00	2.000000	2.000000	0.000000	2.000000
25%	2.800000e+02	365243.000000	475.000000	257.000000	455.000000
50%	5.810000e+02	365243.000000	921.000000	741.000000	1155.000000
75%	1.300000e+03	365243.000000	1825.000000	1735.000000	2418.000000
max	2.922000e+03	365243.000000	365243.000000	365243.000000	365243.000000

Standardizing values for days_decision column by converting them into years by binning, for better understanding of data after replacing the null values with mode.



About 35% applicant apply again after decision on previous loan

```
#days group calculation

bins = [0,1*365,2*365,3*365,4*365,5*365,6*365,7*365,10*365]
slots = ["1","2","3","4","5","6","7","7 above"]
pra['YEARLY_DECISION'] = pd.cut(pra['DAYS_DECISION'],bins,labels=slots)
```

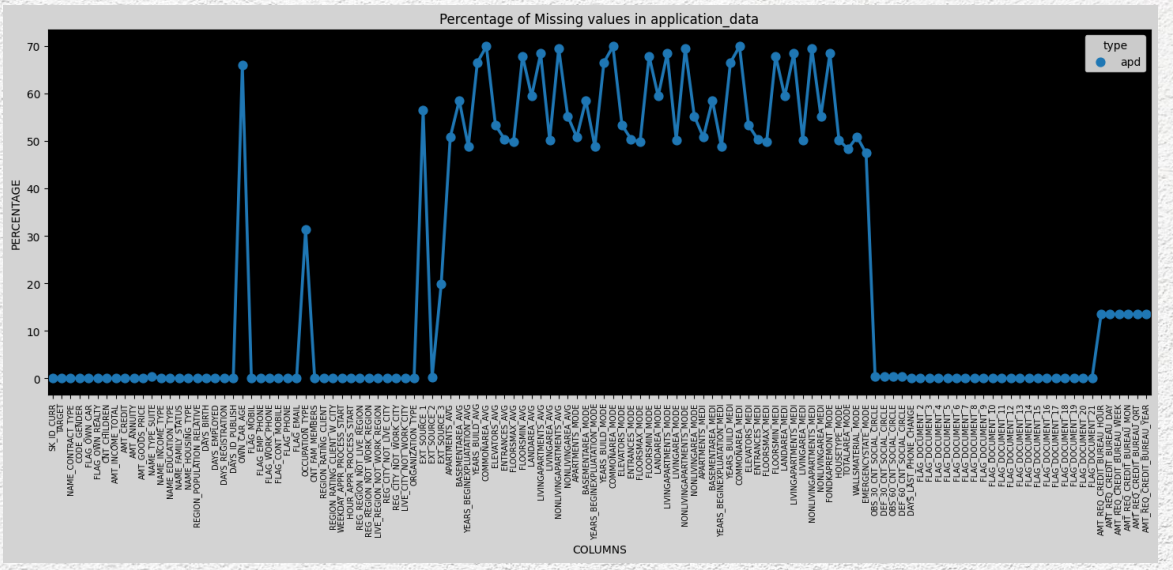


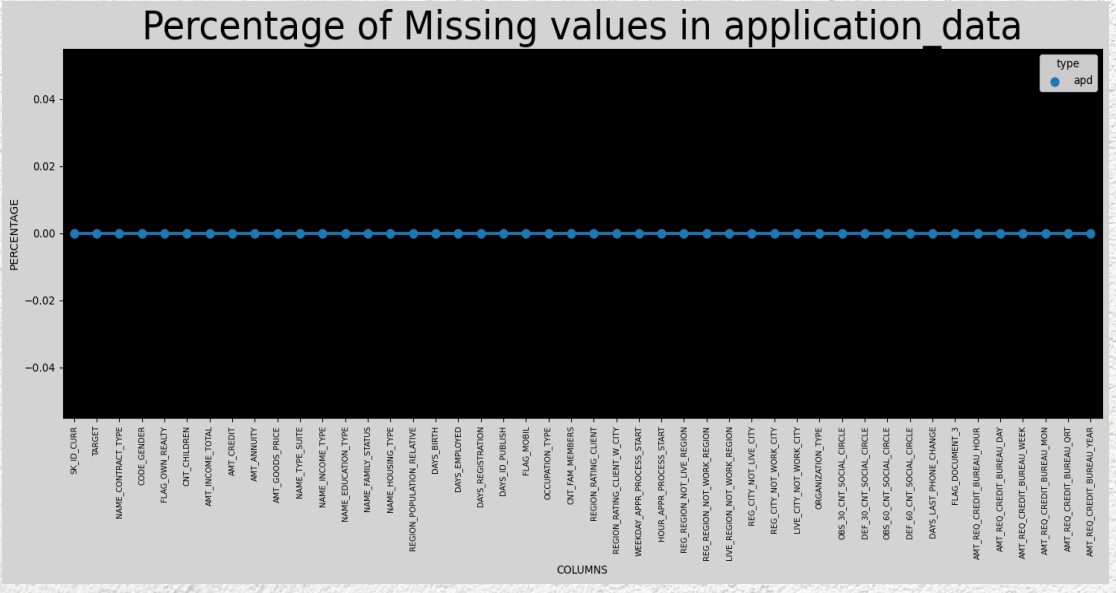
All Null Values have been removed

DATA CLEANING CONCLUSION

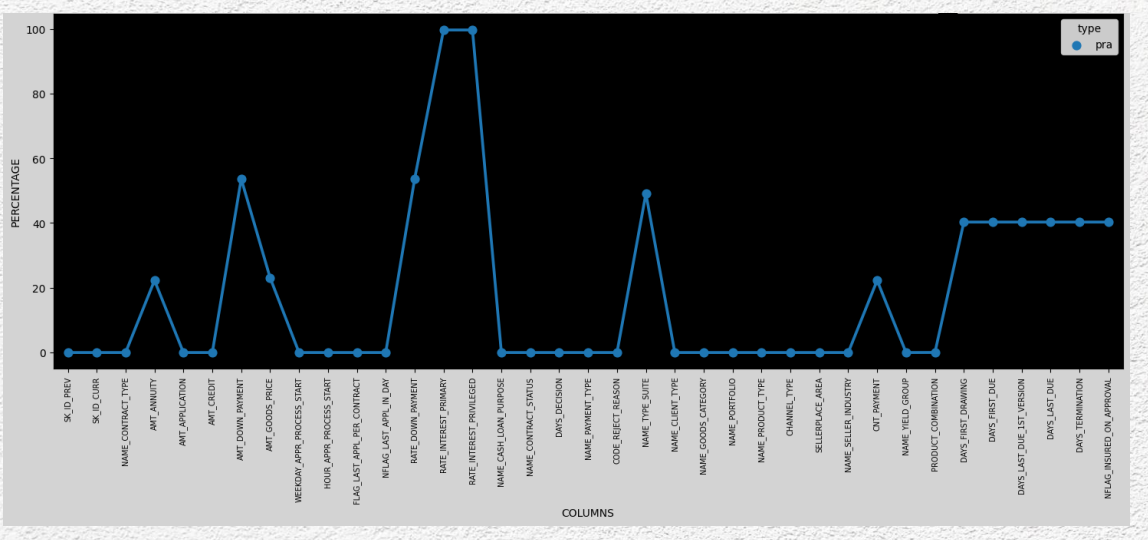
DATASET-1 APPLICATION DATA-> from 122 columns down to 53 columns

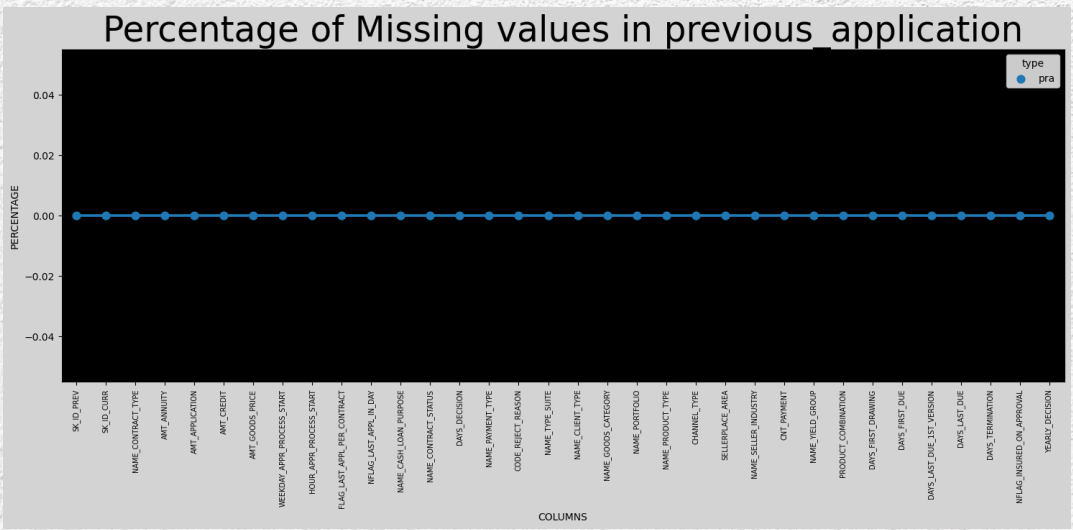
DATASET-2 PREVIOUS APPLICATION from 37 columns down to 34 columns


Percentage of Missing values in application_data
type
apd
PERCENTAGE
COLUMNS


Percentage of Missing values in application_data
type
apd
PERCENTAGE
COLUMNS

DATASET-2 PREVIOUS APPLICATION from 37 columns down to 34 columns





TASK-2-IDENTIFYING OUTLIER

OUTLIER IDENTIFICATION
DATA SET-1 APPLICATION DATA

From describe we could find all the columns those who have high difference between max and 75 percentile and the ones which makes no sense having max value to be so high are captured below:

```
outlier_col = ["CNT_CHILDREN","AMT_INCOME_TOTAL", "AMT_CREDIT", "AMT_ANNUITY", "AMT_GOODS_PRICE",
               "DAYS_BIRTH", "DAYS_EMPLOYED", "DAYS_REGISTRATION", "CNT_FAM_MEMBERS"]
```

The box plot of these columns will reflect if they have outliers or not.

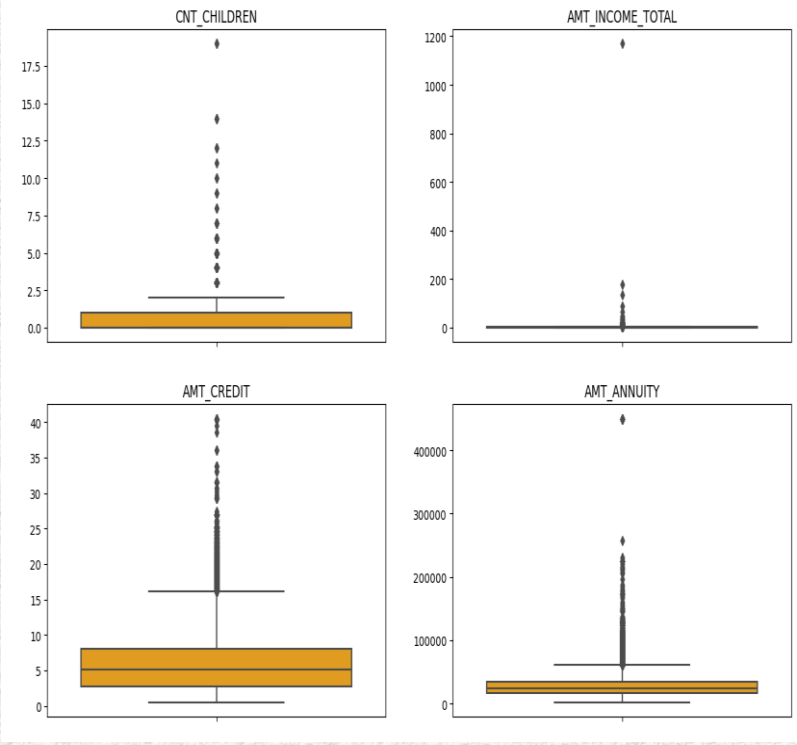
AMT_ANNUITY, AMT_CREDIT, AMT_GOODS_PRICE,CNT_CHILDREN have some number of outliers.

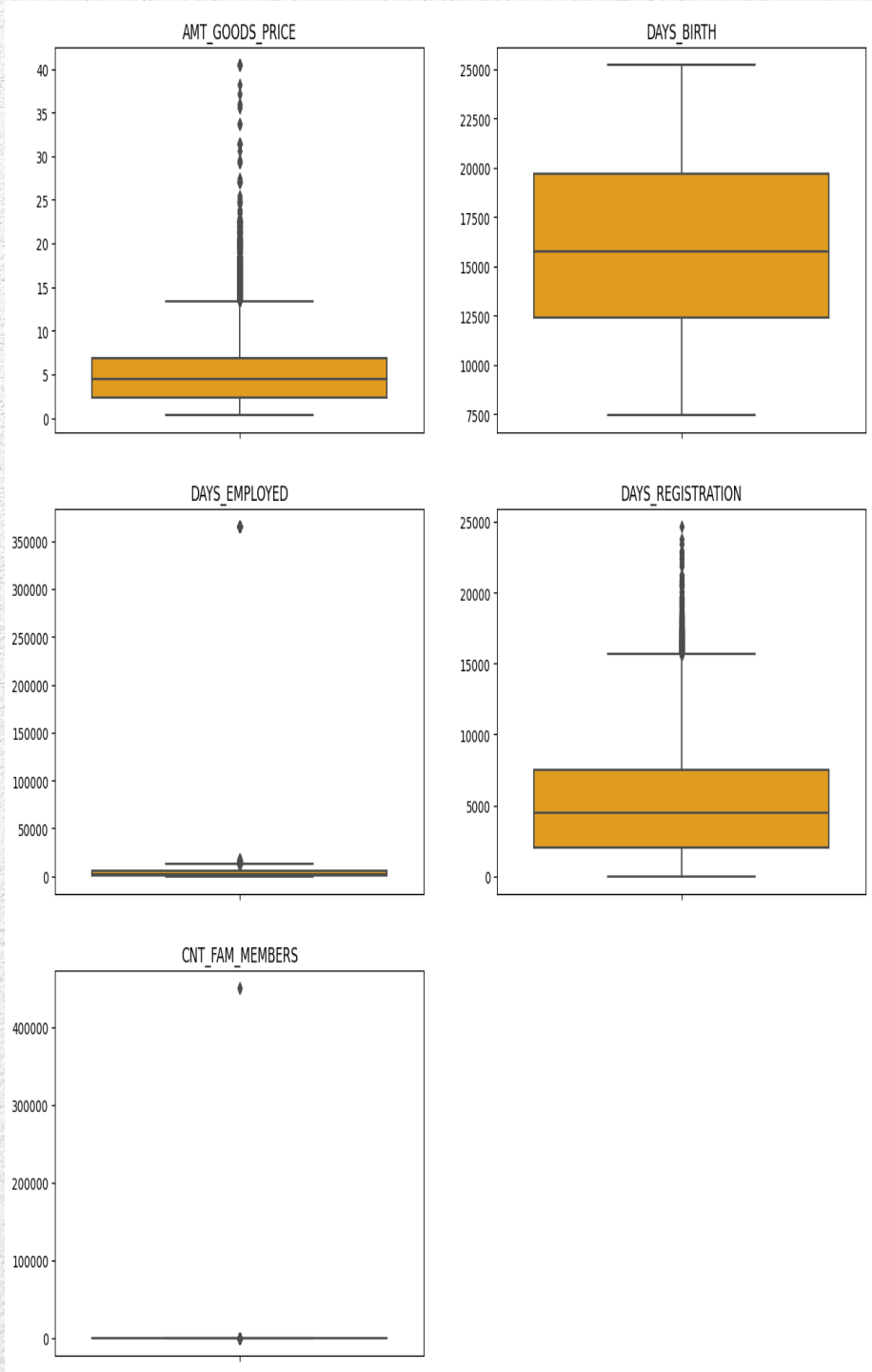
AMT_INCOME_TOTAL has huge number of outliers which indicate that few of the loan applicants have high income when compared to the others.

DAYS_BIRTH has no outliers which means the data available is reliable.

DAYS_EMPLOYED has outlier values around 350000(days) which is around 958 years which is impossible and hence this has to be incorrect entry.

CNT_FAM_MEMBERS has outlier value more than 45000 which is impossible hence it is an incorrect entry




AMT_GOODS_PRICE
0
5
10
15
20
25
30
35
40
DAYS_BIRTH
7500
10000
12500
15000
17500
20000
22500
25000
DAYS_EMPLOYED
0
50000
100000
150000
200000
250000
300000
350000
DAYS_REGISTRATION
0
5000
10000
15000
20000
25000
CNT_FAM_MEMBERS
0
100000
200000
300000
400000

OUTLIER IDENTIFICATION
DATA SET-2 PREVIOUS APPLICATION

From describe we could find all the columns those who have high difference between max and 75 percentile and the ones which makes no sense having max value to be so high are captured below:

```
p_outlier_col = ['AMT_ANNUITY','AMT_APPLICATION','AMT_CREDIT','AMT_GOODS_PRICE',
                 'SELLERPLACE_AREA','DAYS_DECISION','CNT_PAYMENT']
```

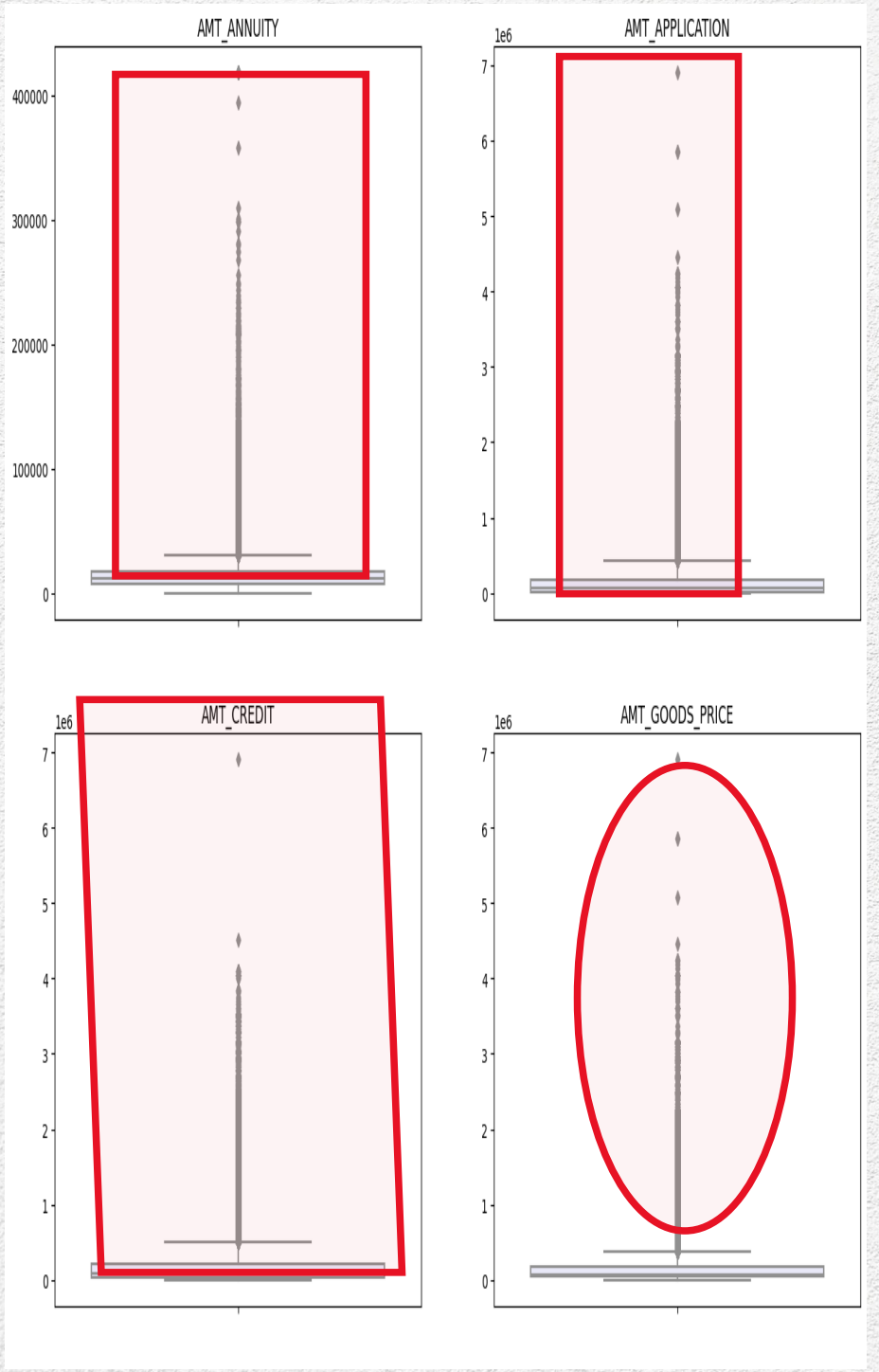
The box plot of these columns will reflect if they have outliers or not.

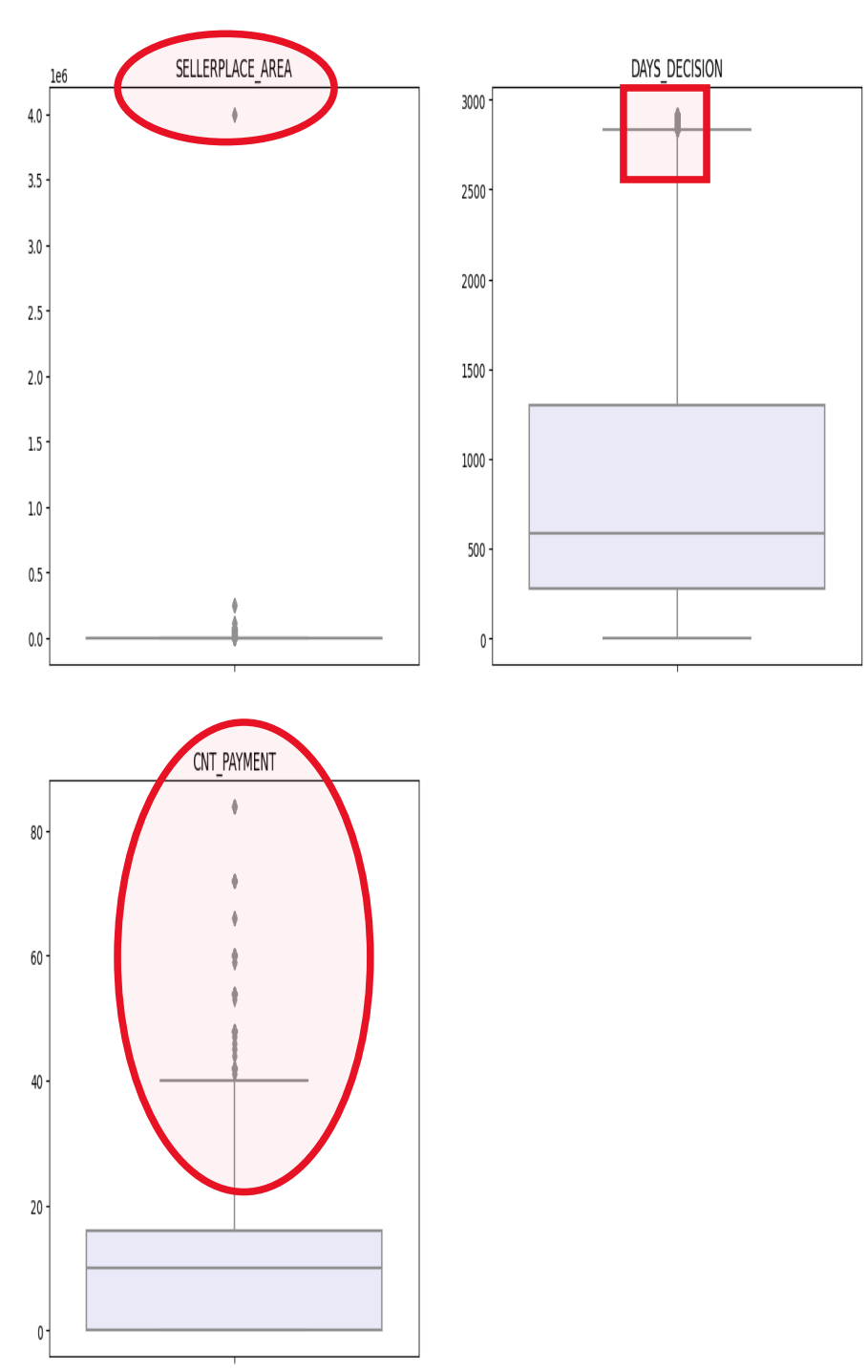
It can be seen that in previous application data

- AMT_ANNUITY, AMT_APPLICATION, AMT_CREDIT, AMT_GOODS_PRICE, SELLERPLACE_AREA have huge number of outliers.

- CNT_PAYMENT has few outlier values.

- DAYS_DECISION has little number of outliers indicating that these previous applications decisions were taken recently.


AMT_ANNUITY
400000
300000
200000
100000
0
AMT_APPLICATION
1e6
7
6
5
4
3
2
1
0
AMT_CREDIT
1e6
7
6
5
4
3
2
1
0
AMT_GOODS_PRICE
1e6
7
6
5
4
3
2
1
0

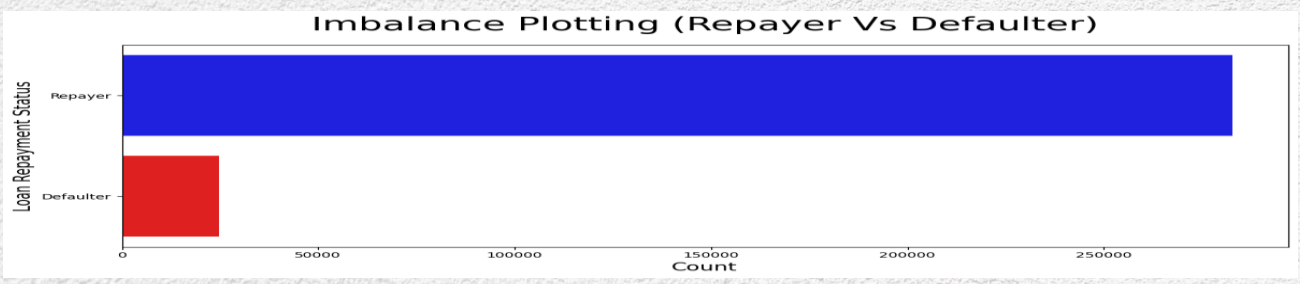

SELLERPLACE_AREA
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
DAYS_DECISION
3000
2500
2000
1500
1000
500
0
CNT_PAYMENT
80
60
40
20
0

TASK-3-REPORTING DATA IMBALANCE

IMBALANCE BETWEEN DEFAULTER AND REPAYER COUNT

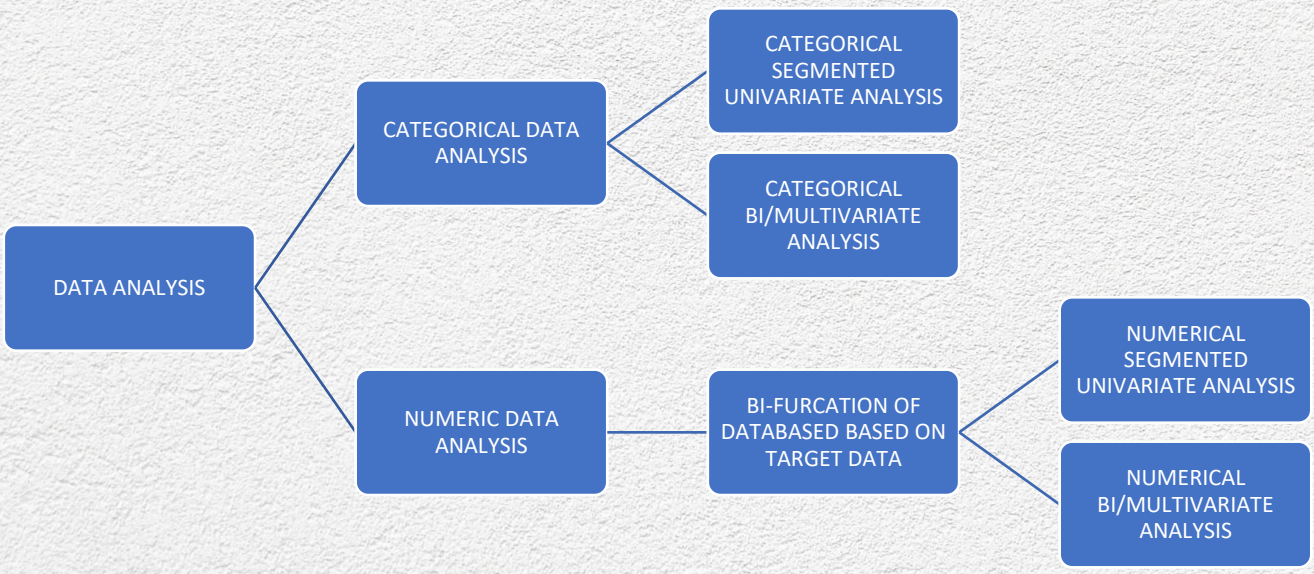
```
No. of reapyers are  282686
No. of defaulters are  24825
```

```
Repayer Percentage is 91.93%
Defaulter Percentage is 8.07%
Imbalance Ratio with respect to Repayer and Defaulter is given: 11.39/1 (approx)
```



TASK-4 REPORTING RESULTS OF UNIVARIATE,SEGMENTED UNIVARAITE AND BIVARIATE ANALYSIS

METHODOLOGY ADOPTED



APPRAOCH ADOPTED

To do the analysis, various functions are created to divide the data sets into categorical and numeric datasets

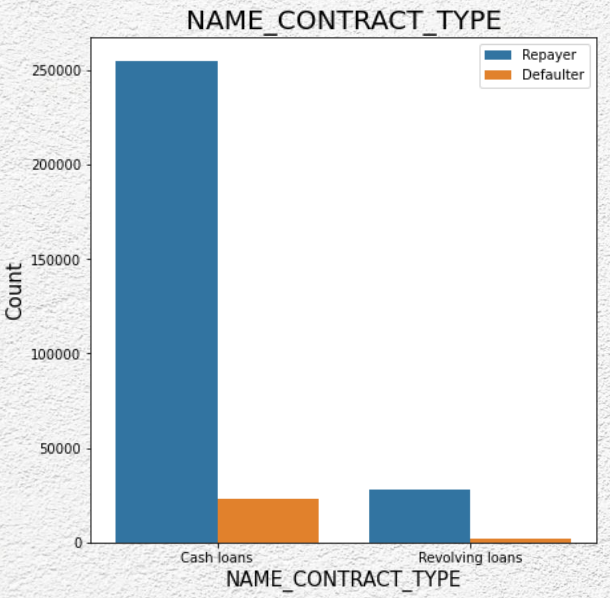
Later new more functions are created to analyse the dataset according to the required analysis (univariate/segmented univariate/ bivariate)

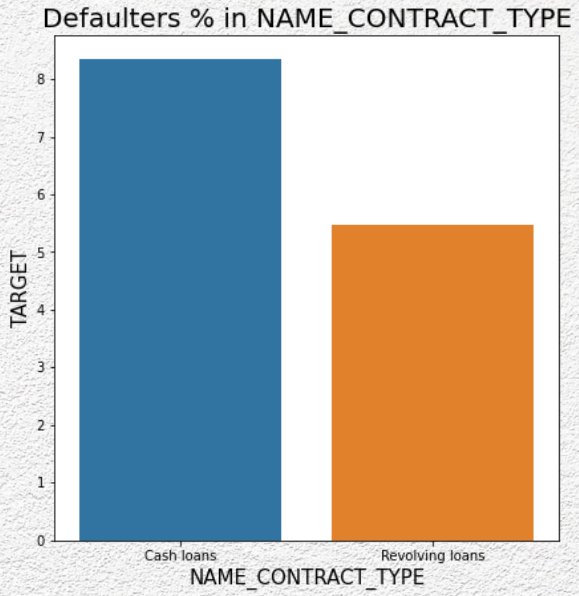
(THE FUNCTIONS CAN BE SEEN IN THE IPYNB ATTATCHED IN TECH STACK USED SLIDE)

CATEGORICAL VARIABLES ANALYSIS

SEGMENTED UNIVARIATE ANALYSIS

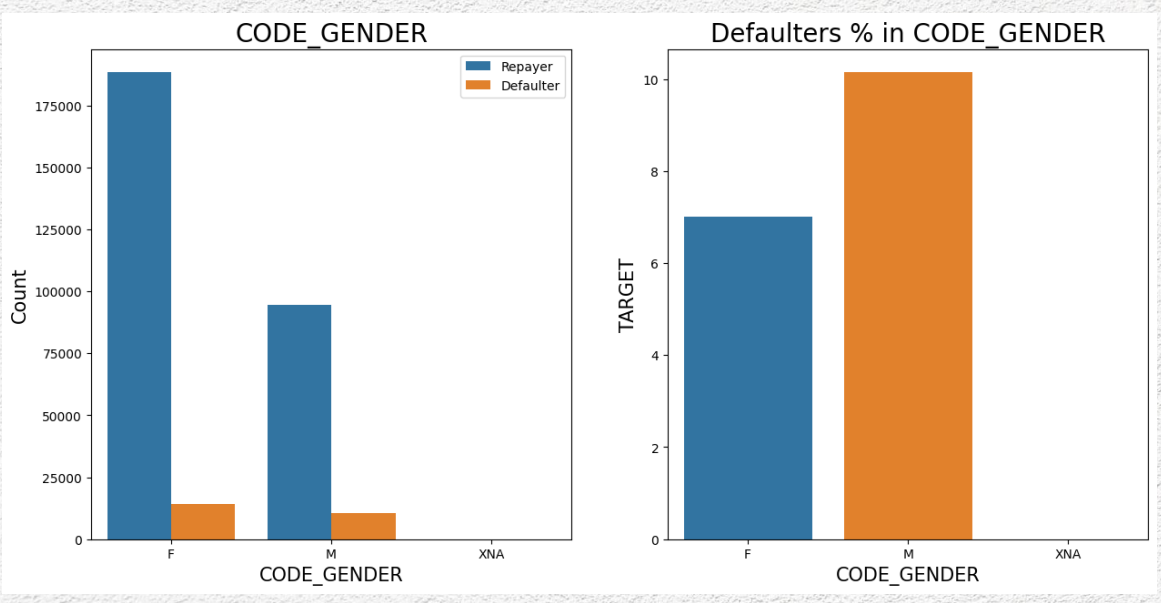
CONTRACT TYPE





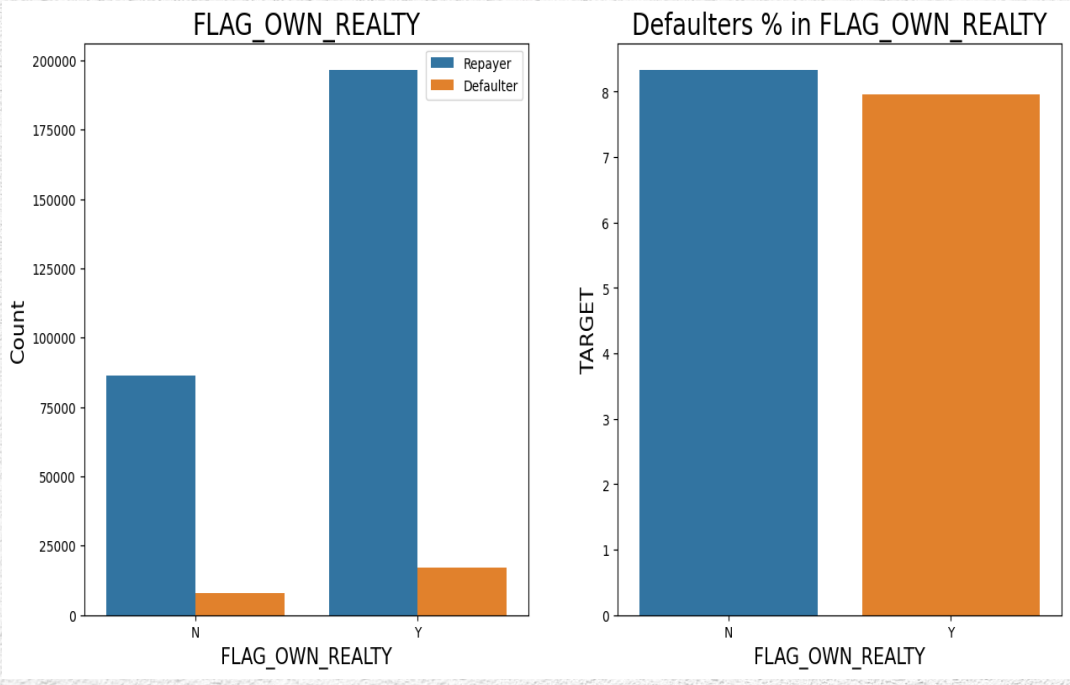
Revolving loans are just a small fraction (10%) from the total number of loans.
Around 8-9% Cash loan applicants and 5-6% Revolving loan applicant are in defaulters.

GENDER WISE LOAN REPAYMENT STATUS



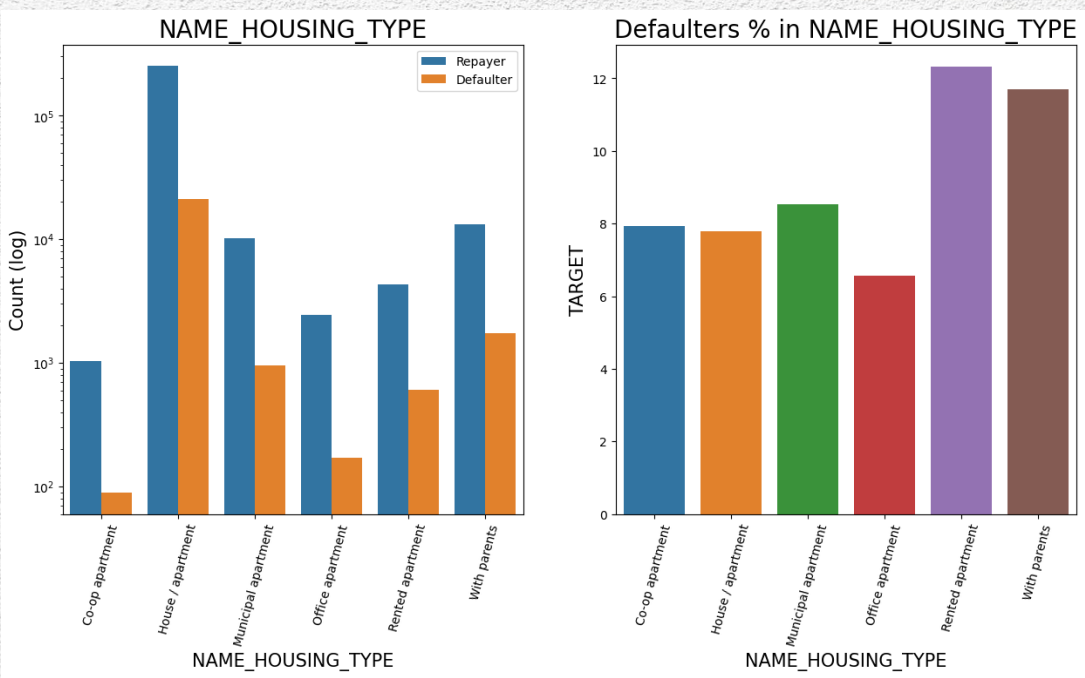
- The number of female clients is almost double the number of male clients.
- Based on the percentage of defaulted credits, males have a higher chance of not returning their loans about 10%, comparing with women about 7%.

REAL ESTATE OWNERSHIP AND LOAN REPAYMENT STATUS



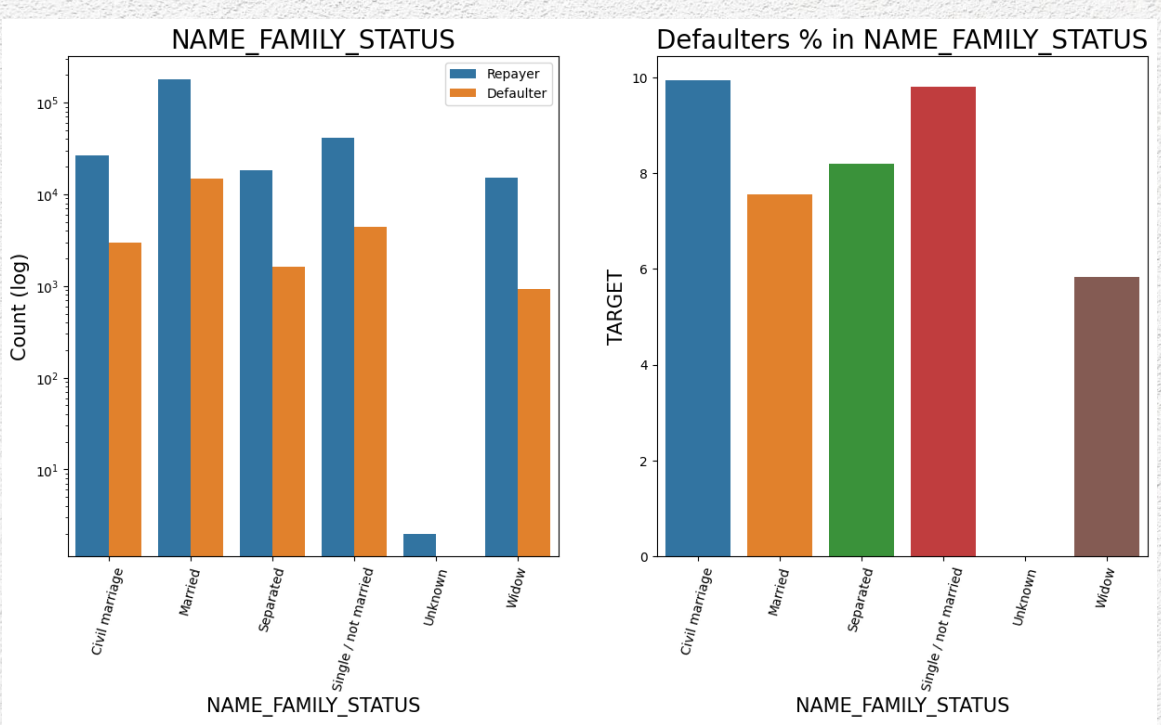
- The clients who own real estate are more than double of the ones that don't own.
- The defaulting rate of both categories are around the same (~8%). Thus we can infer that there is no correlation between owning a reality and defaulting the loan.

HOUSING TYPE AND LOAN REPAYMENT STATUS



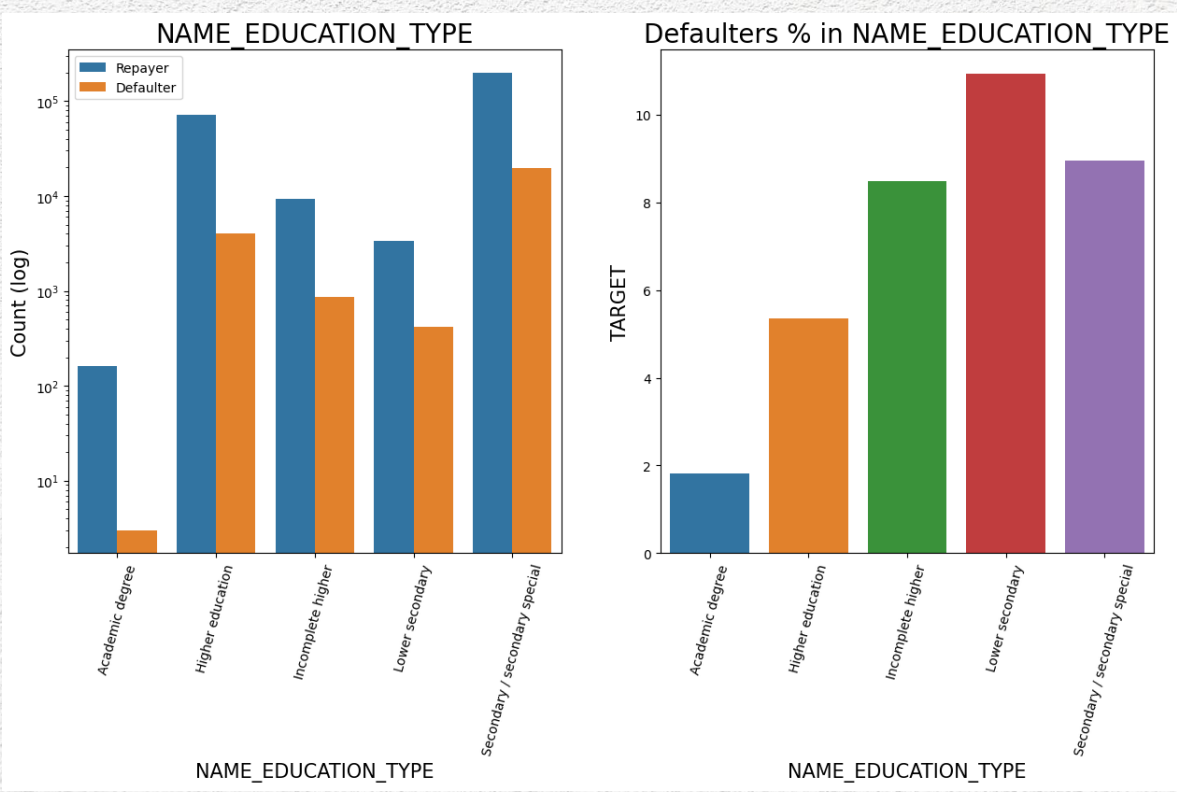
- Majority of people live in House/apartment.
- People living in office apartments have lowest default rate.
- People living with parents (~11.5%) and living in rented apartments(>12%) have higher probability of defaulting.

FAMILY STATUS AND LOAN REPAYMENT STATUS



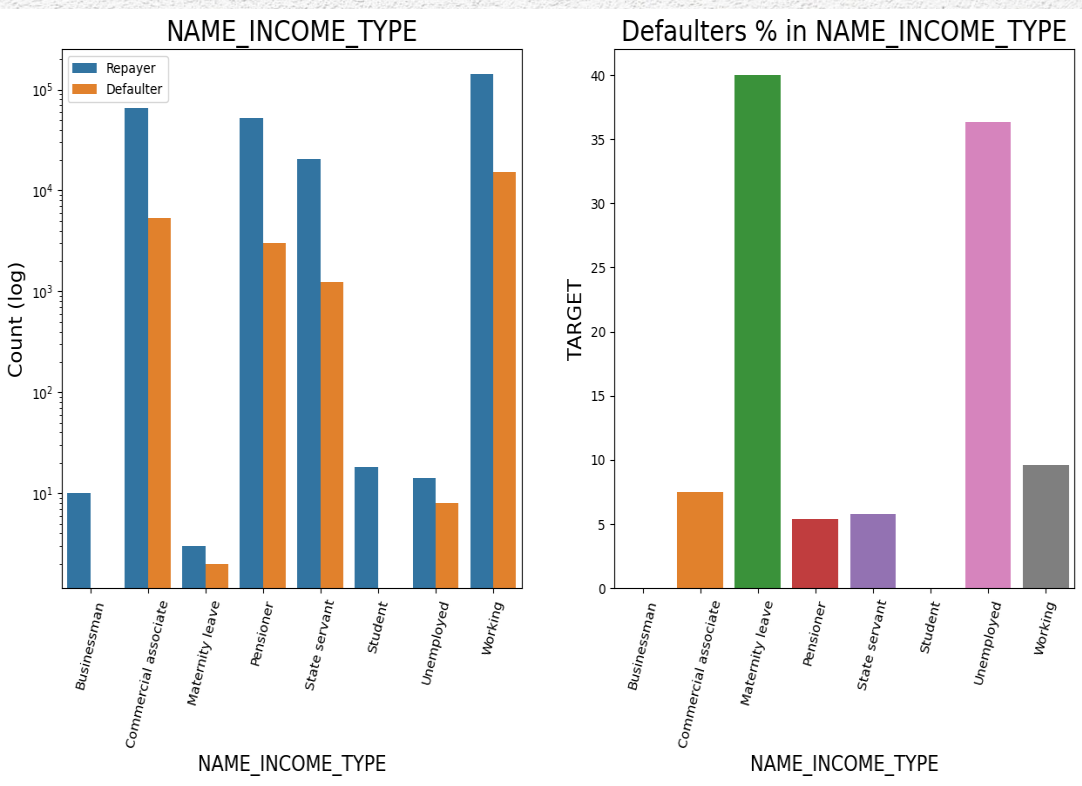
- Most of the people who have taken loan are married, followed by Single/not married and civil marriage.
- In Percentage of defaulters, Civil marriage has the highest percent around (10%) and widow has the lowest around 6% (exception being Unknown).

EDUCATION AND LOAN REPAYMENT STATUS



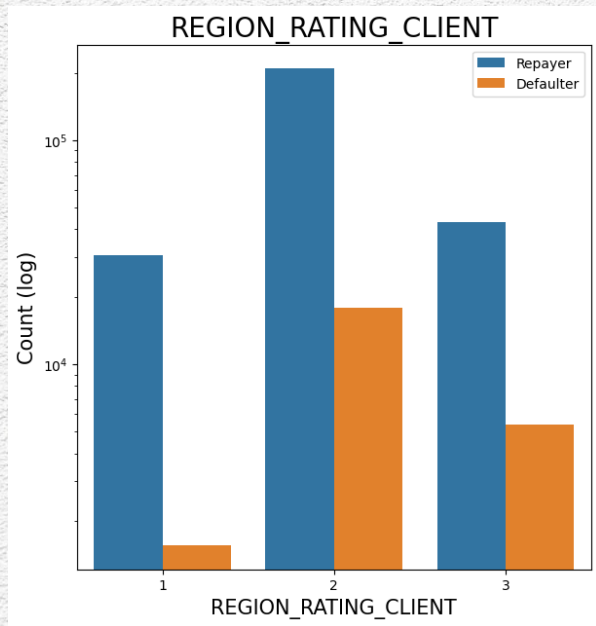
- Majority of clients have Secondary/secondary special education, followed by clients with Higher education.
- Very few clients have an academic degree
- Lower secondary category have highest rate of defaulting around 11%.
- People with Academic degree are least likely to default.

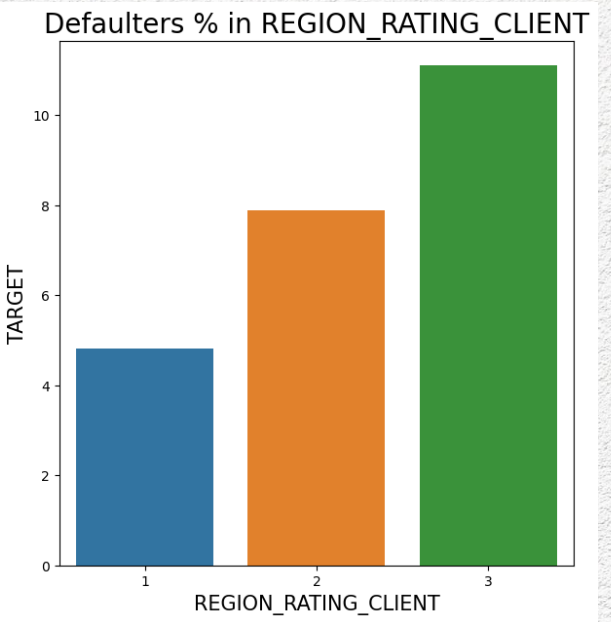
INCOME TYPE AND LOAN REPAYMENT STATUS



- Most of applicants for loans income type is Working, followed by Commercial associate, Pensioner and State servant.
- The applicants who are on Maternity leave have defaulting percentage of 40% which is the highest, followed by Unemployed (37%). The rest under average around 10% defaulters.
- Student and Businessmen though less in numbers, do not have default record. Safest two categories for providing loan.

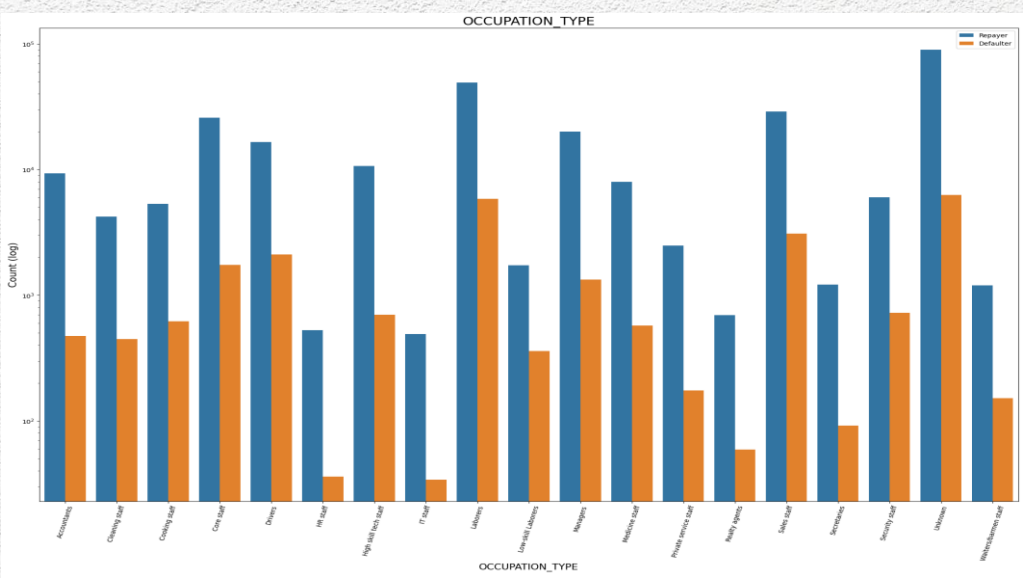
REGION RATING AND LOAN REPAYMENT STATUS

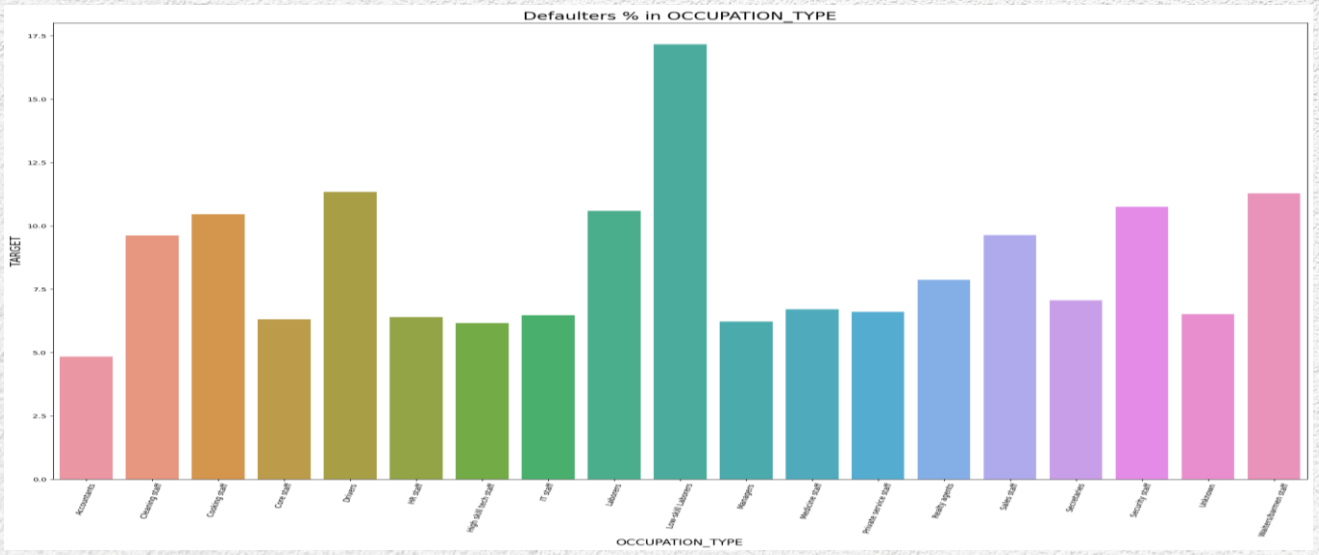




- Most of the applicants are living in Region with Rating 2 place.
- Region Rating 3 has the highest default rate (11%).
- Applicant living in Region_Rating 1 has the lowest probability of defaulting, thus safer for approving loans.

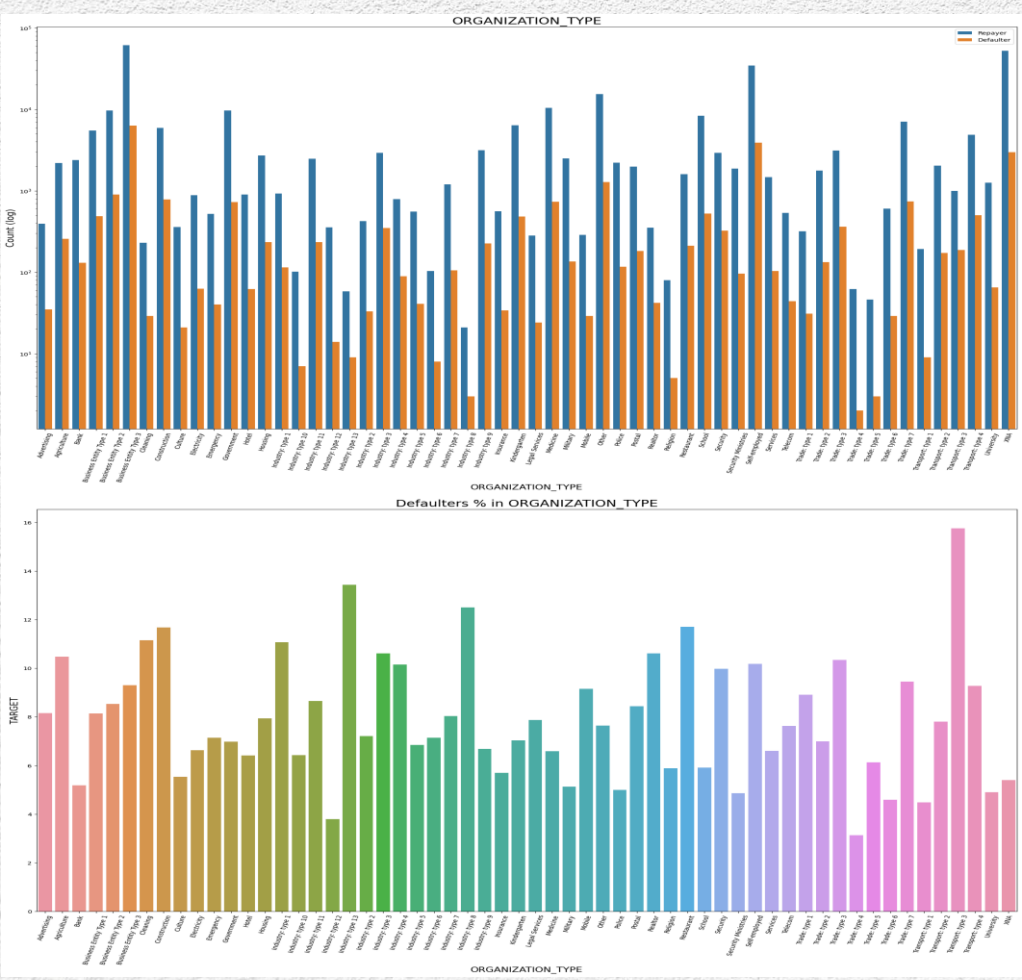
OCCUPATION TYPE AND LOAN REPAYMENY STATUS





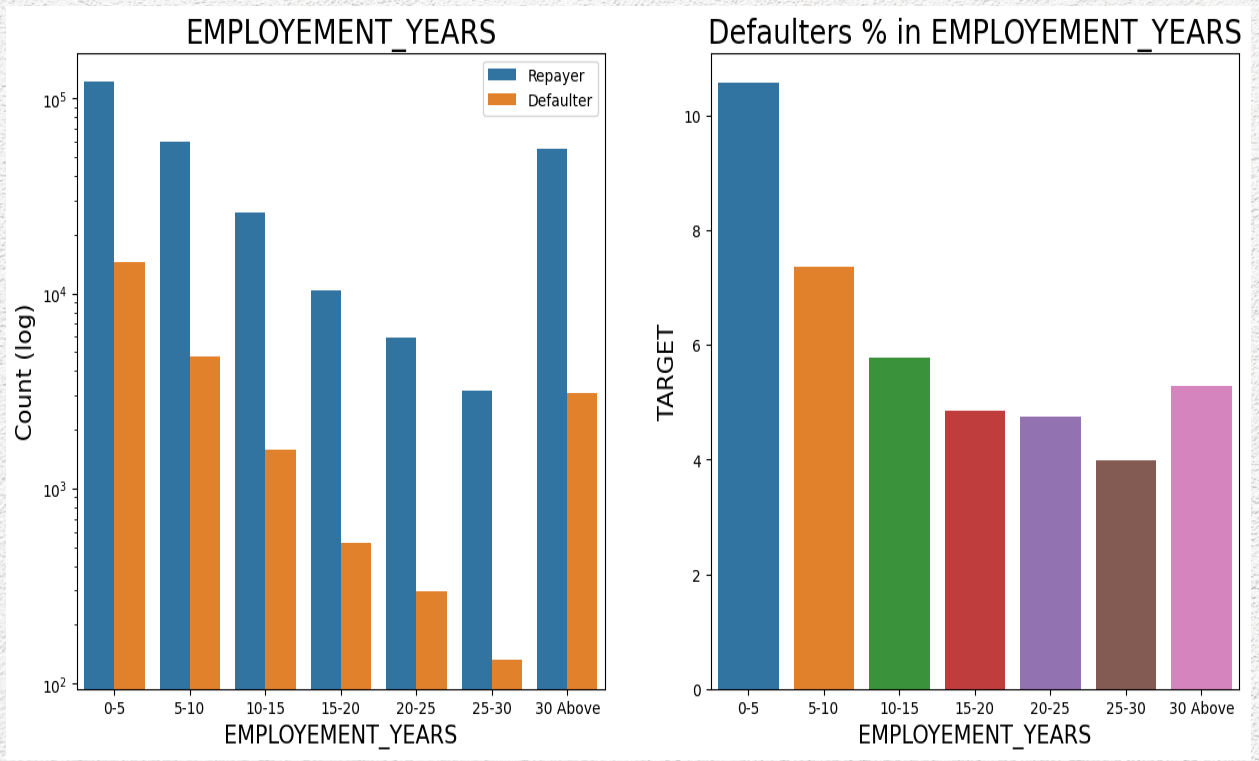
Most of the loans are taken by Laborers, followed by Sales staff.
IT staff are less likely to apply for Loan.
Category with highest percent of defaulters are Low-skill Laborers (above 17%), followed by Drivers and Waiters/barmen staff, Security staff, Laborers and Cooking staff.

ORGANIZATION TYPE AND LOAN REPAYMENT STATUS



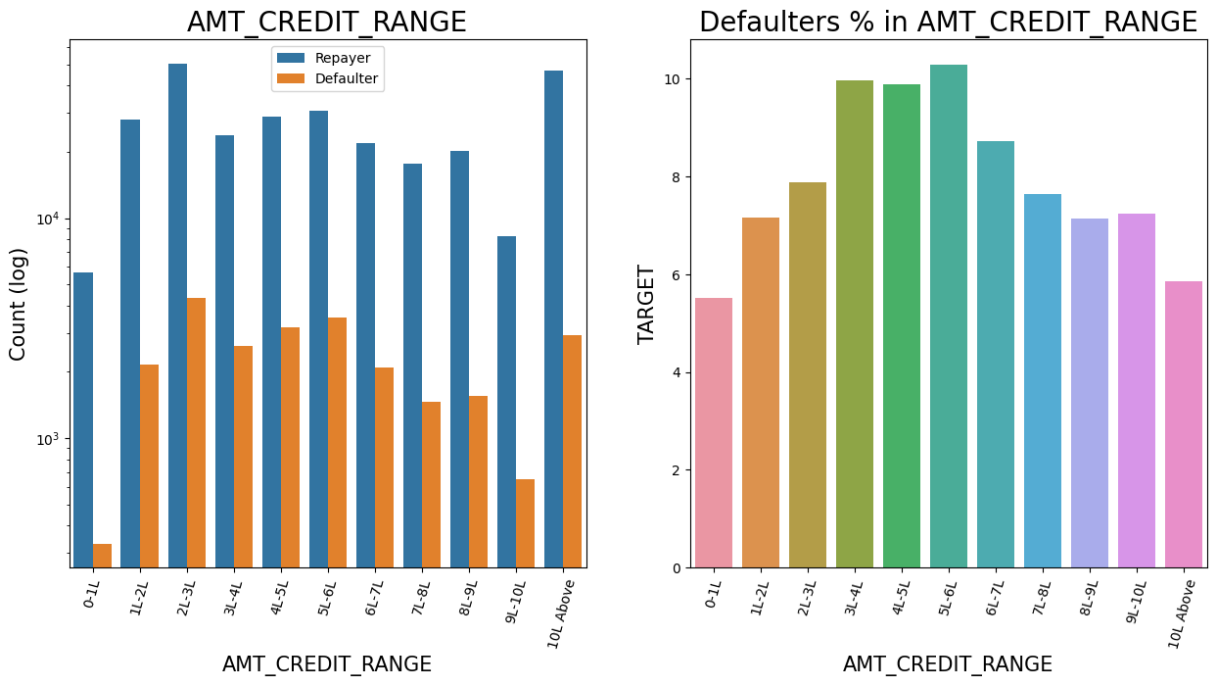
Organizations with highest percent of defaulters are Transport: type 3 (16%), Industry: type 13 (13.5%), Industry: type 8 (12.5%) and Restaurant (less than 12%).
Self employed people have relative high defaulting rate, to be on safer side loan disbursement should be avoided or provide loan with higher interest rate to mitigate the risk of defaulting.
Most of the people application for loan are from Business Entity Type 3.
For a very high number of applications, Organization type information is unavailable(XNA).
It can be seen that following category of organization type has lesser defaulters thus safer for providing loans: Trade Type 4 and 5, Industry type 8.

EMPLOYMENT YEARS AND LOAN REPAYMENT STATUS



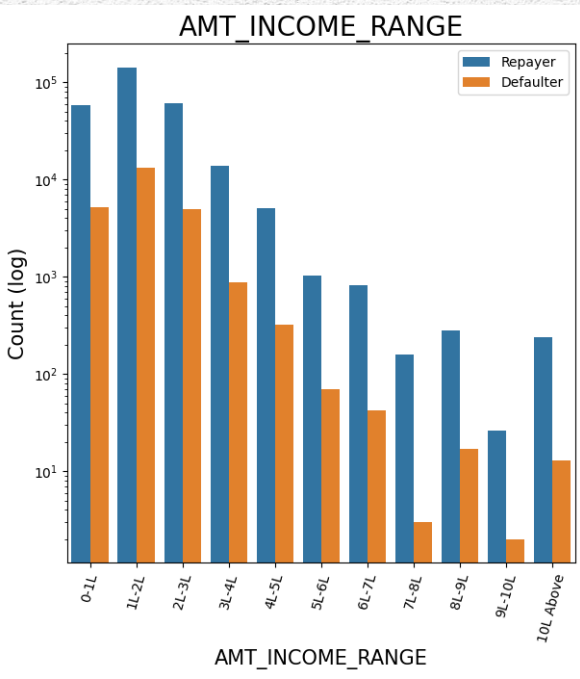
- Majority of the applicants have working experience between 0-5 years. The defaulting rating of this group is also the highest which is around 10%.
- With increase of employment year, defaulting rate is gradually decreasing.
- Those with experience of 25-30 years have lowest default percentage.

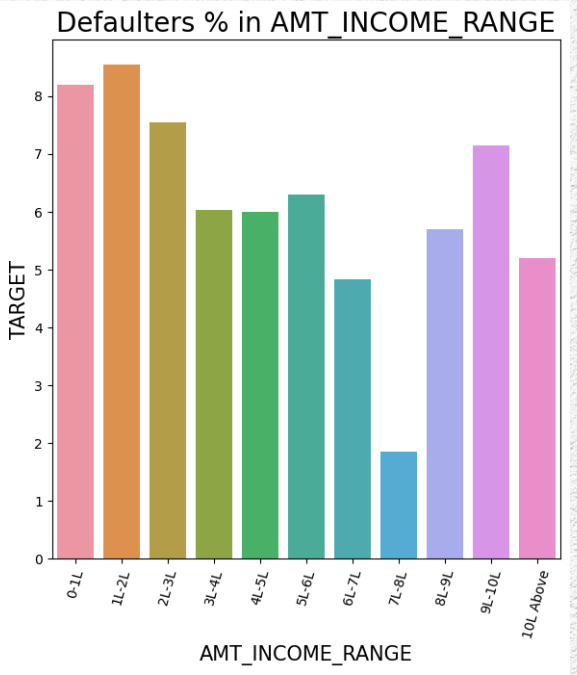
AMOUNT CREDIT AND LOAN REPAYMENT STATUS



- there are high number of applicants have loan in range of 2-3 Lakhs followed by 10 Lakh above range.
- People who get loan for 3-6 Lakhs have most number of defaulters than other loan range.

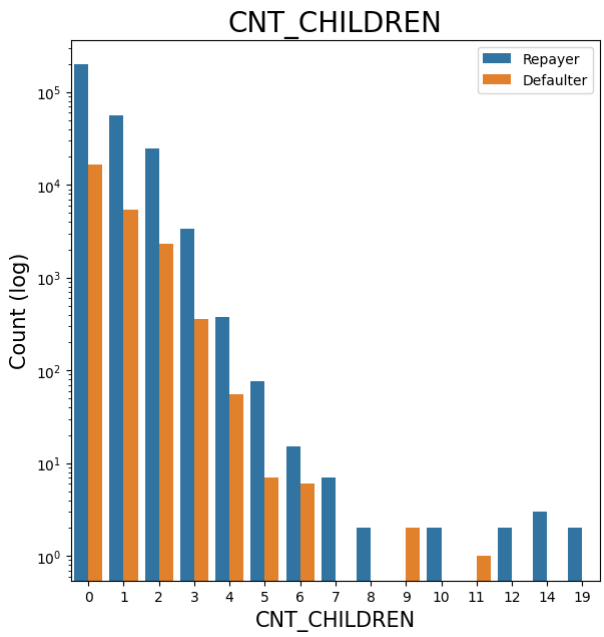
INCOME RANGE AND LOAN REPAYMENT STATUS

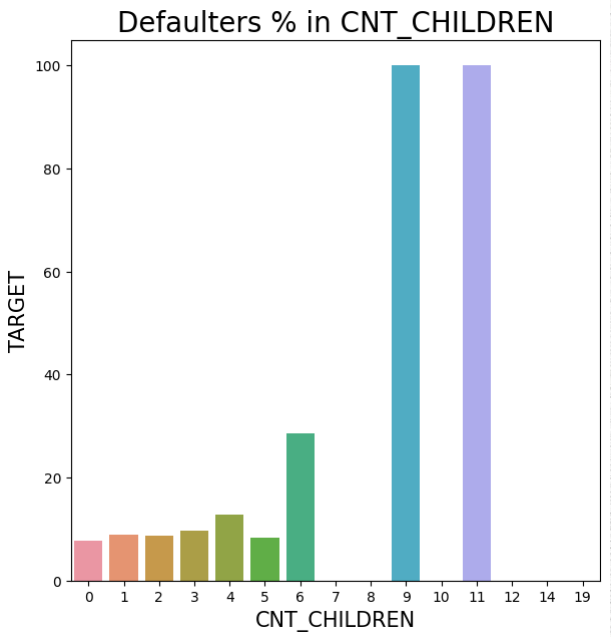




- Majority of the applications have Income total less than 3 Lakhs.
- Application with Income less than 3 Lakhs has high probability of defaulting.
- Applicant with Income 7-8 Lakhs are less likely to default.

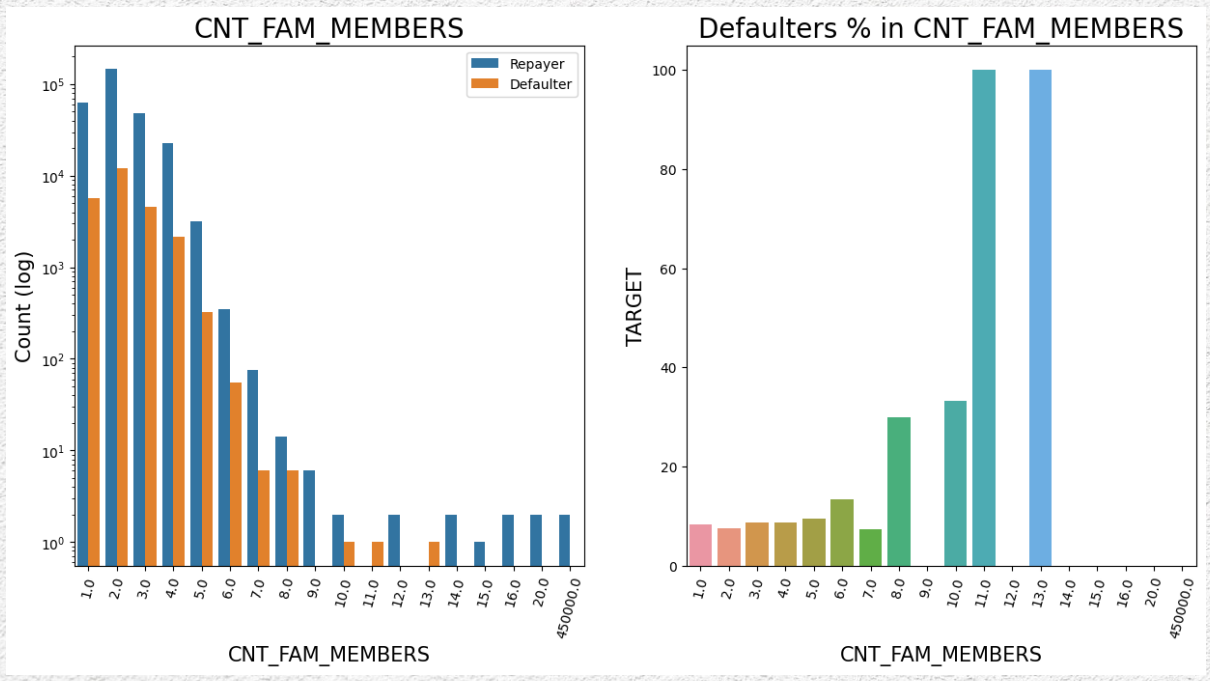
CHILDREN COUNT AND LOAN REPAYMENT STATUS





- Most of the applicants do not have children.
- Very few clients have more than 3 children.
- Client who have more than 4 children has a very high default rate with child count 9 and 11 showing 100% default rate.

COUNT FAMILY MEMBERS AND LOAN REPAYMENT STATUS

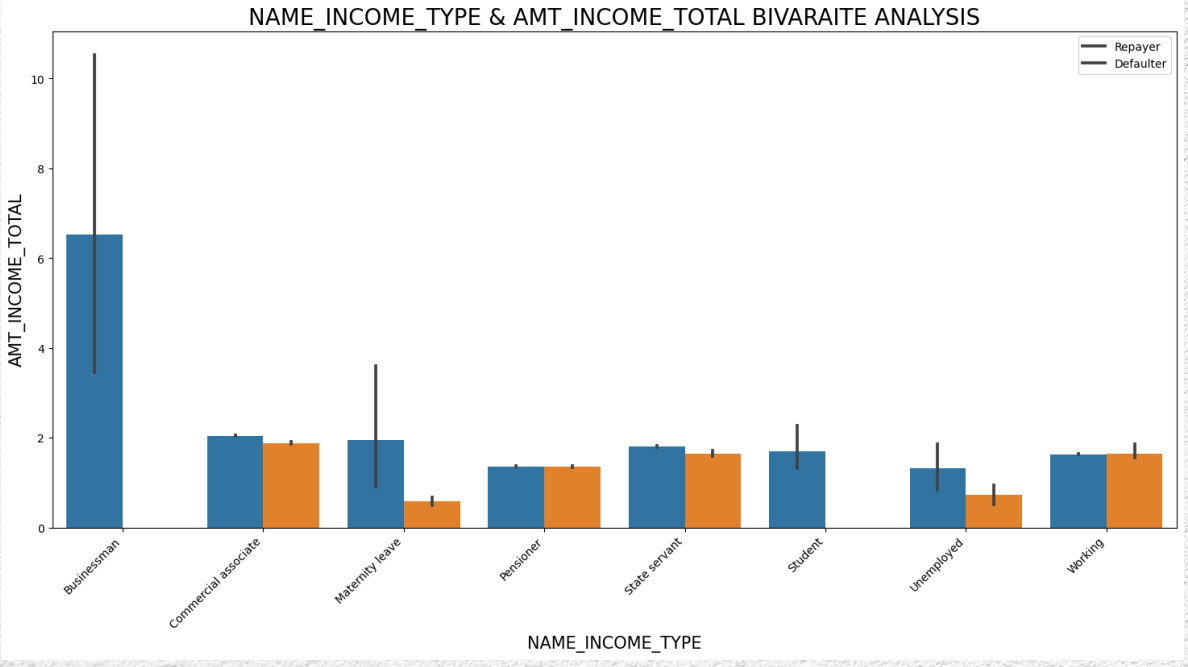


Family member follows the same trend as children where having more family members increases the risk of defaulting.
A Family with 11 & 13 members have 100% default rate.

CATEGORICAL VARIABLES ANALYSIS

BIVARIATE/MULTIVARAITE ANALYSIS

INCOME TYPE AND TOTAL INCOME RELATIONSHIP

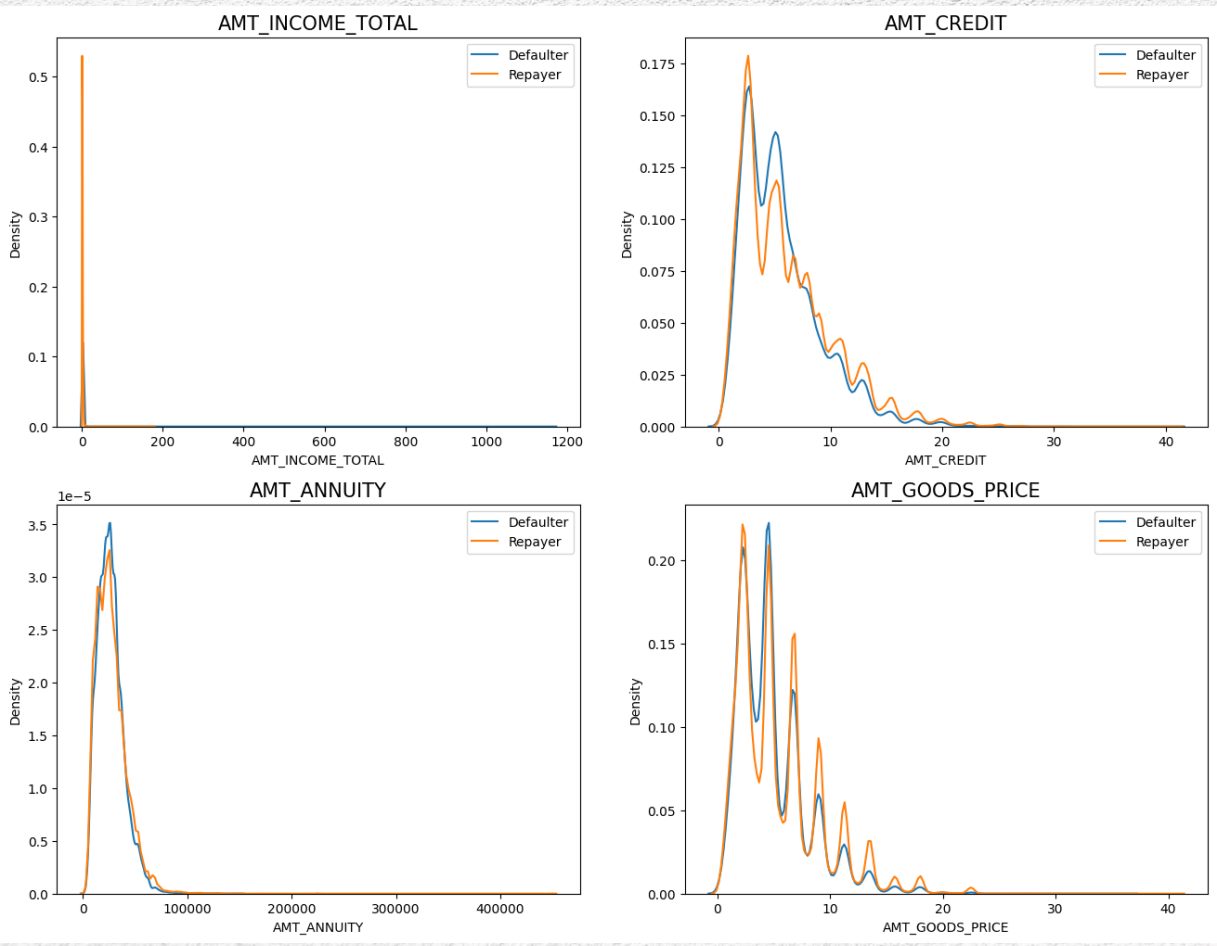


It can be seen that Businessman income is the highest and the estimated range with default 95% confidence level seem to indicate that the income of a Businessman could be in the range of slightly close to 4 lakhs and slightly above 10 lakhs.

NUMERICAL VARIABLES ANALYSIS

UNIVARIATE ANALYSIS

AMOUNT AND LOAN REPAYMENT STATUS

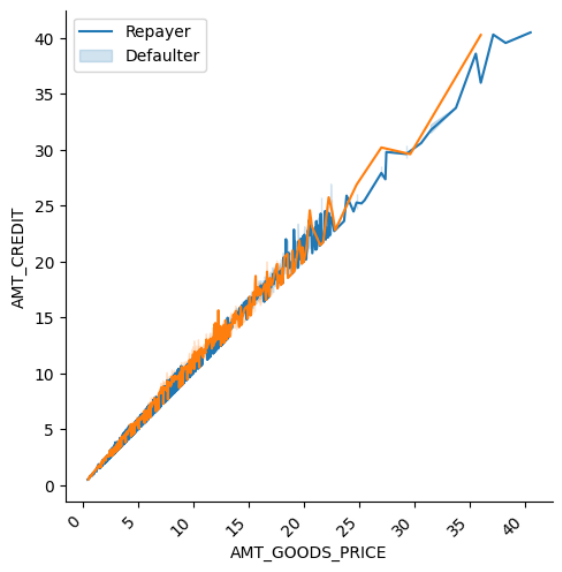


Most no of loans are given for goods price below 10 lakhs.
Most people pay annuity below 50K for the credit loan.
Credit amount of the loan is mostly less then 10 lakhs.
The repayers and defaulters distribution overlap in all the plots and hence we cannot use any of these variables in isolation to make a decision.

NUMERICAL VARIABLES ANALYSIS

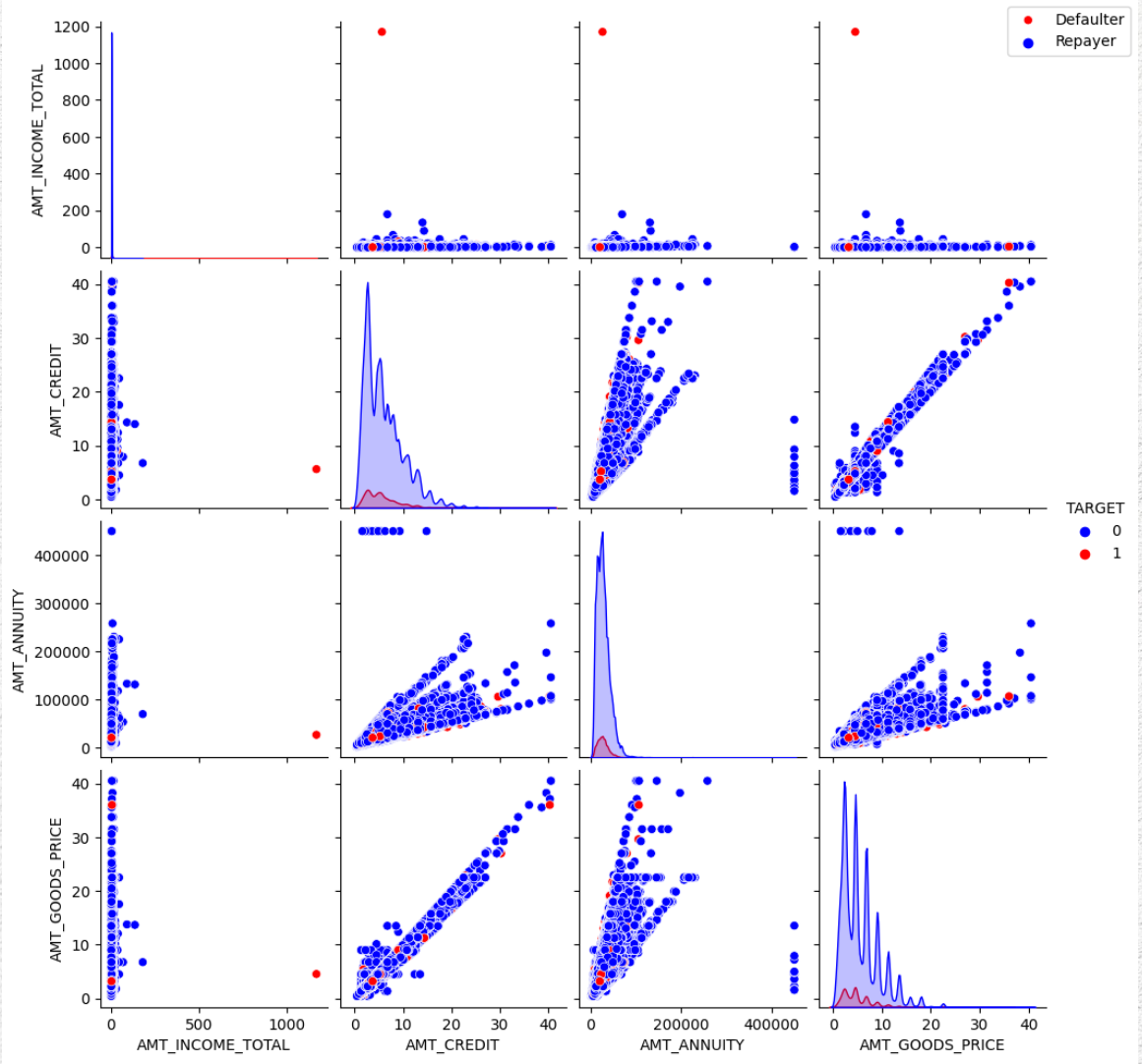
BIVARIATE ANALYSIS

GOOD PRICE AMOUNT AND CREDIT AMOUNT BIVARIATE ANALYSIS



When the credit amount goes beyond 30 Lakhs, there is an increase in defaulters.

LOAN REPAYMENT STATUS BY PLOTTING AGAINST AMOUNT VARIABLES

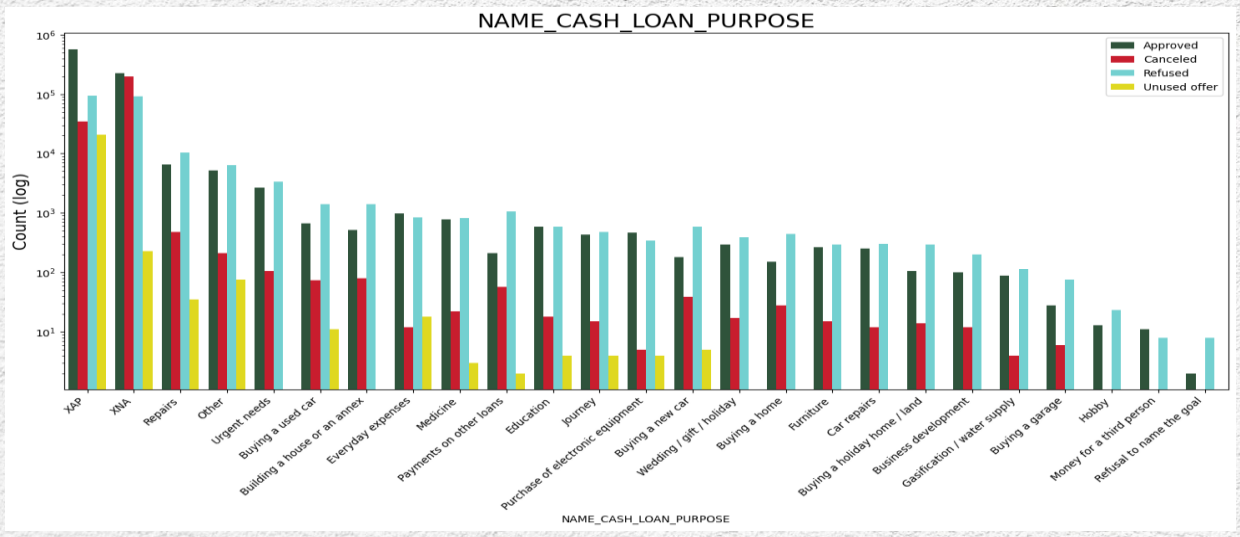


- When Annuity Amount > 15K and Good Price Amount > 20 Lakhs, there is a lesser chance of defaulters.
- Loan Amount(AMT_CREDIT) and Goods price(AMT_GOODS_PRICE) are highly correlated as based on the scatterplot where most of the data are consolidated in form of a line.
- There are very less defaulters for AMT_CREDIT >20 Lakhs.

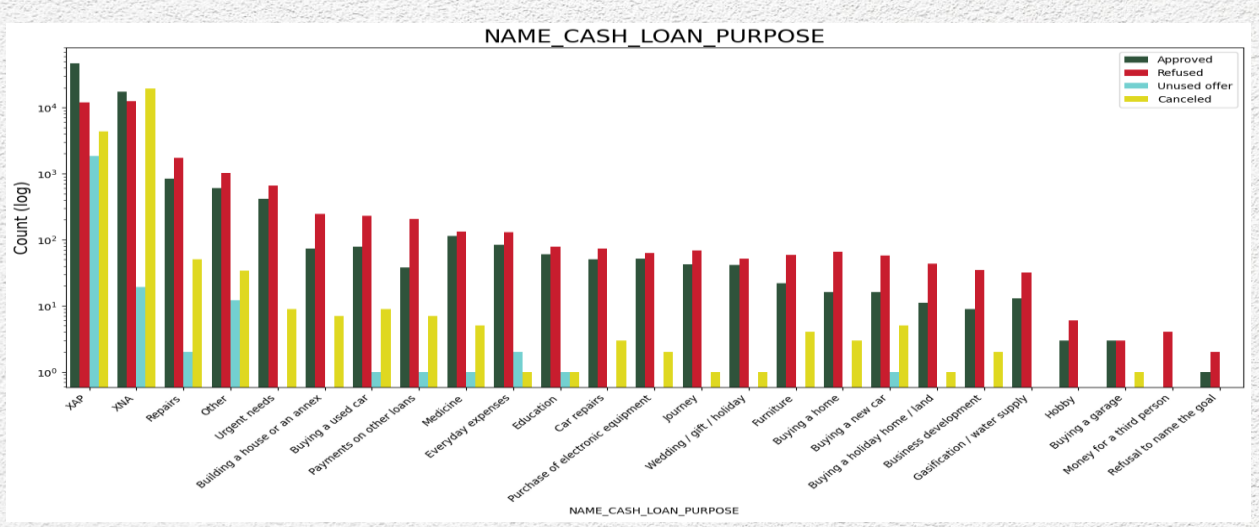
DATA SET-3 VARAIABLE ANALYSIS->MERGEED DATA SETS APPLICATION DATA ANDPREVIOUS APPLICATION

UNIVARAITE AND BIVARIATE ANALYSIS

CONTRACT STATUS AND REPAYMENT OF LOANS



CONTRACT STATUS AND REPAYMENT OF LOANS

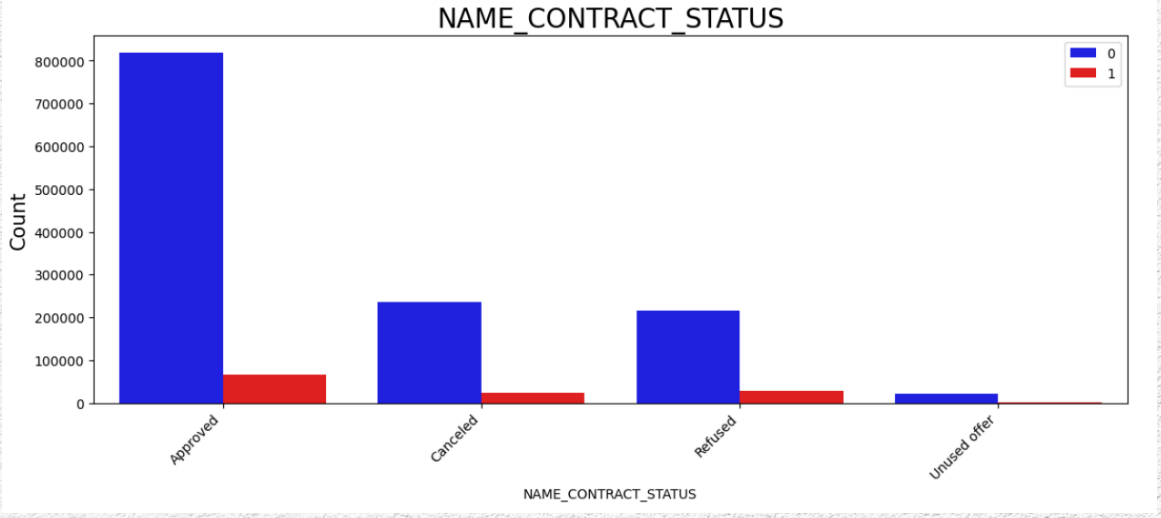


CONTRACT STATUS AND REPAYMENT OF LOANS

FROM THE PERIOUS GRAPHS THE FOLLOWING CAN BE INFERRED

- Loan purpose has high number of unknown values (XAP, XNA).
- Loan taken for the purpose of Repairs looks to have highest default rate.
- Huge number application have been rejected by bank or refused by client which are applied for Repair or Other. from this we can infer that repair is considered high risk by bank. Also, either they are rejected or bank offers loan on high interest rate which is not feasible by the clients and they refuse the loan.

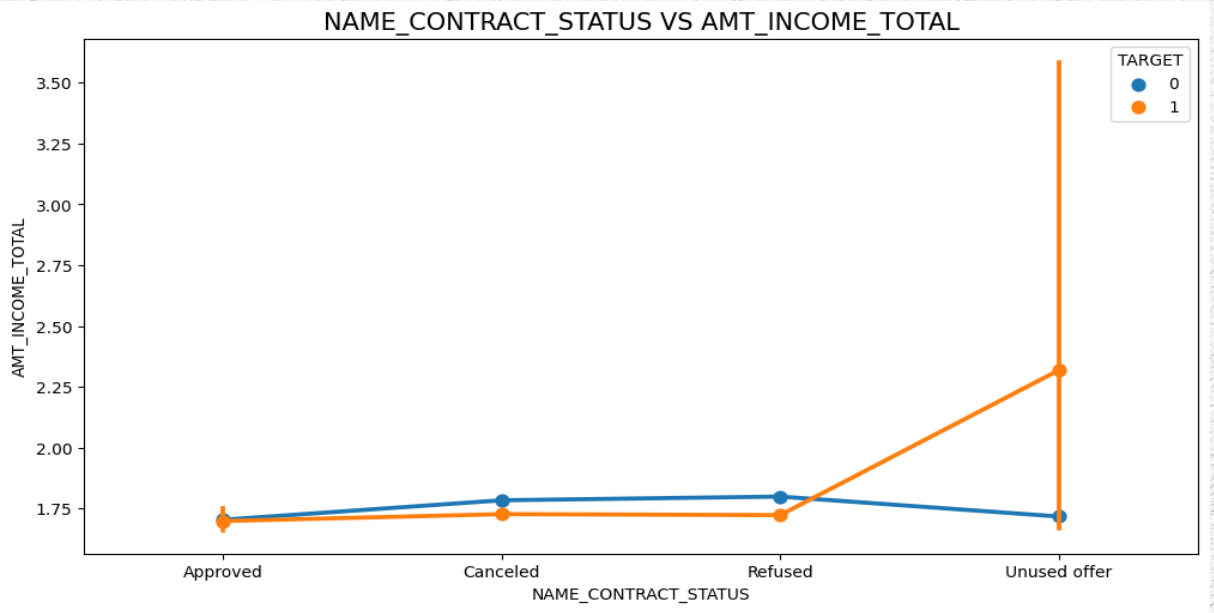
NAME CONTRACT STATUS AND LOAN REPAYMENT STATUS



•90% of the previously cancelled client have actually repaid the loan. Revising the interest rates would increase business opportunity for these clients.
•88% of the clients who have been previously refused a loan has paid back the loan in current case.
•Refusal reason should be recorded for further analysis as these clients could turn into potential repaying customer.

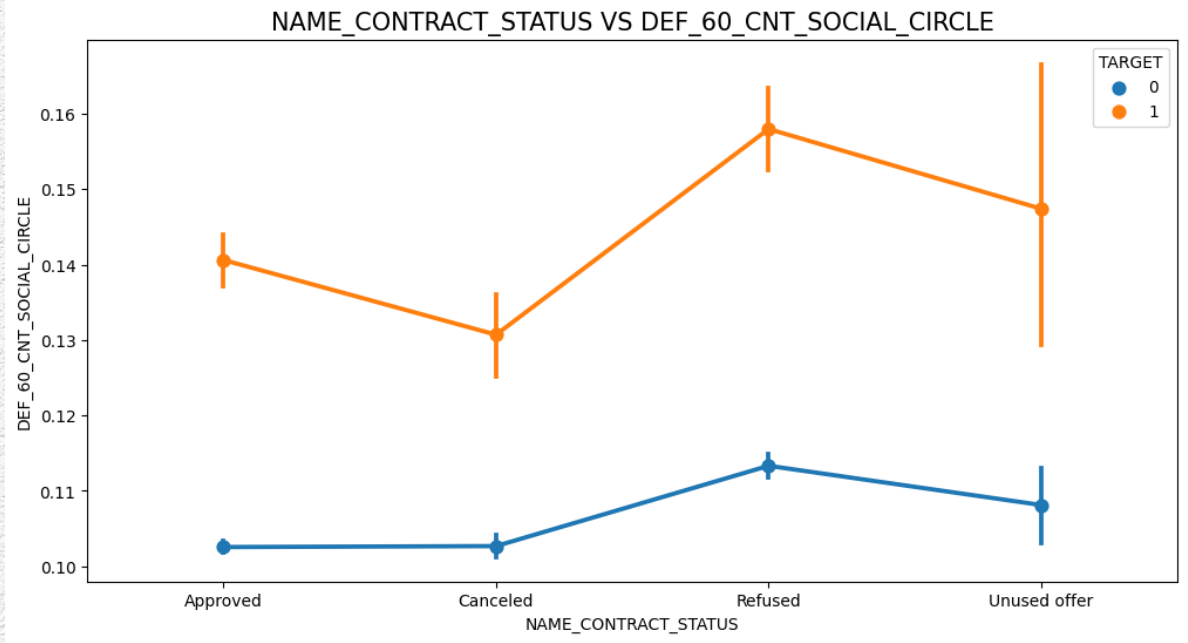
		Counts	Percentage
NAME_CONTRACT_STATUS	TARGET		
Approved	0	818856	92.41%
	1	67243	7.59%
Canceled	0	235641	90.83%
	1	23800	9.17%
Refused	0	215952	88.0%
	1	29438	12.0%
Unused offer	0	20892	91.75%
	1	1879	8.25%

NAME CONTRACT STATUS AND AMOUNT INCOME BIVARIATE ANALYSIS



The point plot show that the people who have not used offer earlier have defaulted even when there average income is higher than others.

CONTRACT STATUS & SOCIAL CIRCLE BIVARAITE ANALYSIS



Clients who have average of 0.13 or higher their DEF_60_CNT_SOCIAL_CIRCLE score tend to default more and thus analysing client's social circle could help in disbursement of the loan.

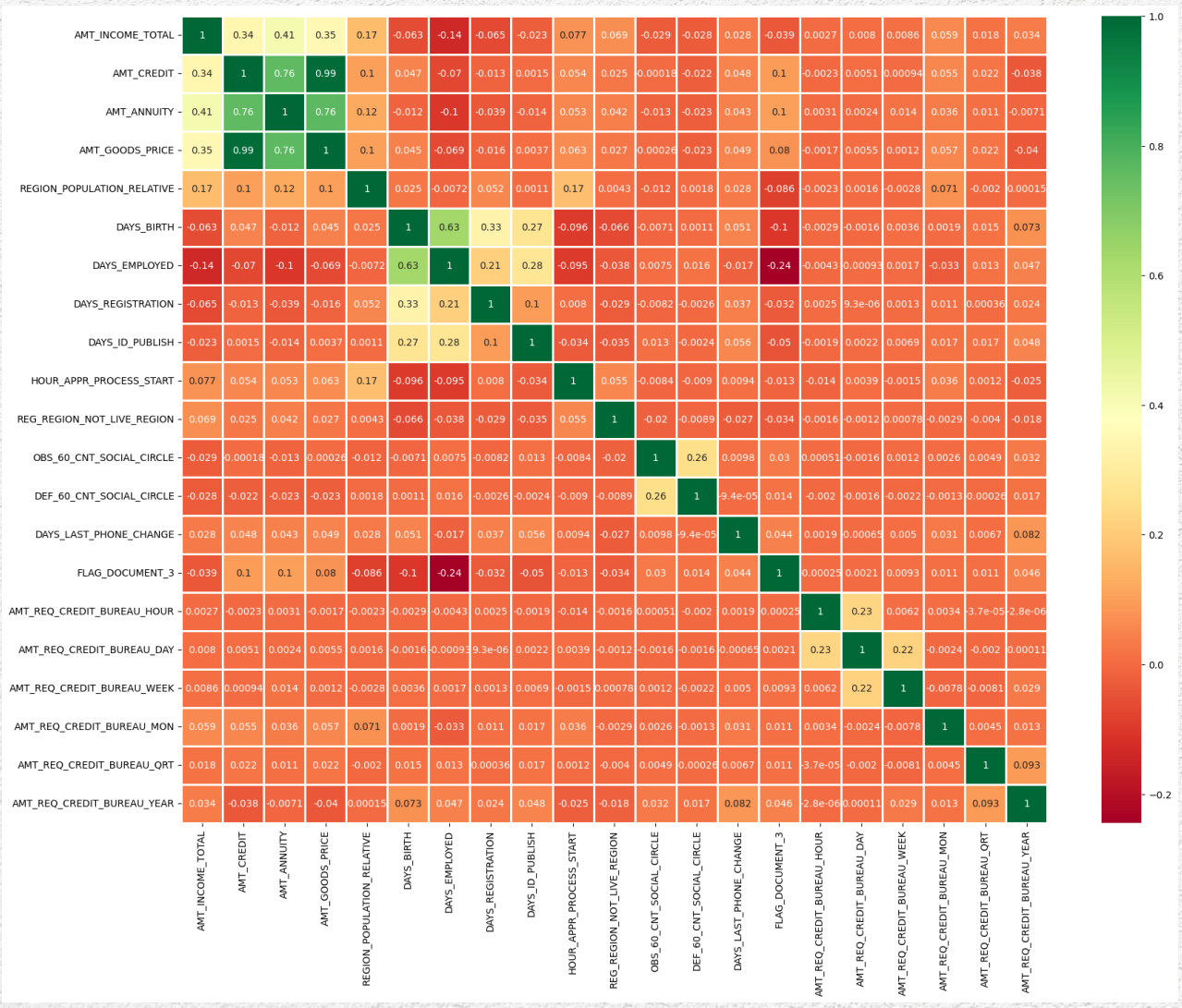
TASK-5 CORRELATEION BETWEEN VARIABLES AND REPAYMENT/DEFAULT

CORRELATION BETWEEN VARIABLES WHEN LOAN REPAID

	VAR1	VAR2	Correlation
64	AMT_GOODS_PRICE	AMT_CREDIT	0.987022
65	AMT_GOODS_PRICE	AMT_ANNUITY	0.762953
43	AMT_ANNUITY	AMT_CREDIT	0.757916
131	DAYS_EMPLOYED	DAYS_BIRTH	0.626114
42	AMT_ANNUITY	AMT_INCOME_TOTAL	0.411929
63	AMT_GOODS_PRICE	AMT_INCOME_TOTAL	0.349426
21	AMT_CREDIT	AMT_INCOME_TOTAL	0.342799
152	DAYS_REGISTRATION	DAYS_BIRTH	0.333151
174	DAYS_ID_PUBLISH	DAYS_EMPLOYED	0.276663
173	DAYS_ID_PUBLISH	DAYS_BIRTH	0.271314

1. Credit amount is highly correlated with:
- Goods Price Amount
- Loan Annuity
- Total Income

2. We can also see that repayers have high correlation in number of days employed.


AMT_INCOME_TOTAL 1 0.34 0.41 0.35 0.17 -0.063 -0.14 -0.065 -0.023 0.077 0.069 -0.029 -0.028 0.028 -0.039 0.0027 0.008 0.0086 0.059 0.018 0.034
AMT_CREDIT 0.34 1 0.76 0.99 0.1 0.047 -0.07 -0.013 0.0015 0.054 0.025 -0.00018 -0.022 0.048 0.1 -0.0023 0.0051 0.00094 0.055 0.022 -0.038
AMT_ANNUITY 0.41 0.76 1 0.76 0.12 -0.012 -0.1 -0.039 -0.014 0.053 0.042 -0.013 -0.023 0.043 0.1 0.0031 0.0024 0.014 0.036 0.011 -0.0071
AMT_GOODS_PRICE 0.35 0.99 0.76 1 0.1 0.045 -0.069 -0.016 0.0037 0.063 0.027 -0.00026 -0.023 0.049 0.08 -0.0017 0.0055 0.0012 0.057 0.022 -0.04
REGION_POPULATION_RELATIVE 0.17 0.1 0.12 0.1 1 0.025 -0.0072 0.052 0.0011 0.17 0.0043 -0.012 0.0018 0.028 -0.086 -0.0023 0.0016 -0.0028 0.071 -0.002 0.00015
DAYS_BIRTH -0.063 0.047 -0.012 0.045 0.025 1 0.63 0.33 0.27 -0.096 -0.066 -0.0071 0.0011 0.051 -0.1 -0.0029 -0.0016 0.0036 0.0019 0.015 0.073
DAYS_EMPLOYED -0.14 -0.07 -0.1 -0.069 -0.0072 0.63 1 0.21 0.28 -0.095 -0.038 0.0075 0.016 -0.017 -0.24 -0.0043 -0.00093 0.0017 -0.033 0.013 0.047
DAYS_REGISTRATION -0.065 -0.013 -0.039 -0.016 0.052 0.33 0.21 1 0.1 0.008 -0.029 -0.0082 -0.0026 0.037 -0.032 0.0025 9.3e-06 0.0013 0.011 0.00036 0.024
DAYS_ID_PUBLISH -0.023 0.0015 -0.014 0.0037 0.0011 0.27 0.28 0.1 1 -0.034 -0.035 0.013 -0.0024 0.056 -0.05 -0.0019 0.0022 0.0069 0.017 0.017 0.048
HOUR_APPR_PROCESS_START 0.077 0.054 0.053 0.063 0.17 -0.096 -0.095 0.008 -0.034 1 0.055 -0.0084 -0.009 0.0094 -0.013 -0.014 0.0039 -0.0015 0.036 0.0012 -0.025
REG_REGION_NOT_LIVE_REGION 0.069 0.025 0.042 0.027 0.0043 -0.066 -0.038 -0.029 -0.035 0.055 1 -0.02 -0.0089 -0.027 -0.034 -0.0016 -0.0012 0.00078 -0.0029 -0.004 -0.018
OBS_60_CNT_SOCIAL_CIRCLE -0.029 0.00018 -0.013 0.00026 -0.012 -0.0071 0.0075 -0.0082 0.013 -0.0084 -0.02 1 0.26 0.0098 0.03 0.00051 -0.0016 0.0012 0.0026 0.0049 0.032
DEF_60_CNT_SOCIAL_CIRCLE -0.028 -0.022 -0.023 -0.023 0.0018 0.0011 0.016 -0.0026 -0.0024 -0.009 -0.0089 0.26 1 -9.4e-05 0.014 -0.002 -0.0016 -0.0022 -0.0013 0.00026 0.017
DAYS_LAST_PHONE_CHANGE 0.028 0.048 0.043 0.049 0.028 0.051 -0.017 0.037 0.056 0.0094 -0.027 0.0098 -9.4e-05 1 0.044 0.0019 0.00065 0.005 0.031 0.0067 0.082
FLAG_DOCUMENT_3 -0.039 0.1 0.1 0.08 -0.086 -0.1 -0.24 -0.032 -0.05 -0.013 -0.034 0.03 0.014 0.044 1 0.00025 0.0021 0.0093 0.011 0.011 0.046
AMT_REQ_CREDIT_BUREAU_HOUR 0.0027 -0.0023 0.0031 -0.0017 -0.0023 -0.0029 -0.0043 0.0025 -0.0019 -0.014 -0.0016 0.00051 -0.002 0.0019 0.00025 1 0.23 0.0062 0.0034 3.7e-05 -2.8e-06
AMT_REQ_CREDIT_BUREAU_DAY 0.008 0.0051 0.0024 0.0055 0.0016 -0.0016 -0.00093 9.3e-06 0.0022 0.0039 -0.0012 -0.0016 -0.0016 0.00065 0.0021 0.23 1 0.22 -0.0024 -0.002 0.00011
AMT_REQ_CREDIT_BUREAU_WEEK 0.0086 0.00094 0.014 0.0012 -0.0028 0.0036 0.0017 0.0013 0.0069 -0.0015 0.00078 0.0012 -0.0022 0.005 0.0093 0.0062 0.22 1 -0.0078 -0.0081 0.029
AMT_REQ_CREDIT_BUREAU_MON 0.059 0.055 0.036 0.057 0.071 0.0019 -0.033 0.011 0.017 0.036 -0.0029 0.0026 -0.0013 0.031 0.011 0.0034 -0.0024 -0.0078 1 0.0045 0.013
AMT_REQ_CREDIT_BUREAU_QRT 0.018 0.022 0.011 0.022 -0.002 0.015 0.013 0.00036 0.017 0.0012 -0.004 0.0049 -0.00026 0.0067 0.011 -3.7e-05 -0.002 -0.0081 0.0045 1 0.093
AMT_REQ_CREDIT_BUREAU_YEAR 0.034 -0.038 -0.0071 -0.04 0.00015 0.073 0.047 0.024 0.048 -0.025 -0.018 0.032 0.017 0.082 0.046 -2.8e-06 0.00011 0.029 0.013 0.093 1
AMT_INCOME_TOTAL
AMT_CREDIT
AMT_ANNUITY
AMT_GOODS_PRICE
REGION_POPULATION_RELATIVE
DAYS_BIRTH
DAYS_EMPLOYED
DAYS_REGISTRATION
DAYS_ID_PUBLISH
HOUR_APPR_PROCESS_START
OBS_60_CNT_SOCIAL_CIRCLE
DEF_60_CNT_SOCIAL_CIRCLE
DAYS_LAST_PHONE_CHANGE
FLAG_DOCUMENT_3
AMT_REQ_CREDIT_BUREAU_HOUR
AMT_REQ_CREDIT_BUREAU_DAY
AMT_REQ_CREDIT_BUREAU_WEEK
AMT_REQ_CREDIT_BUREAU_MON
AMT_REQ_CREDIT_BUREAU_QRT
AMT_REQ_CREDIT_BUREAU_YEAR
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

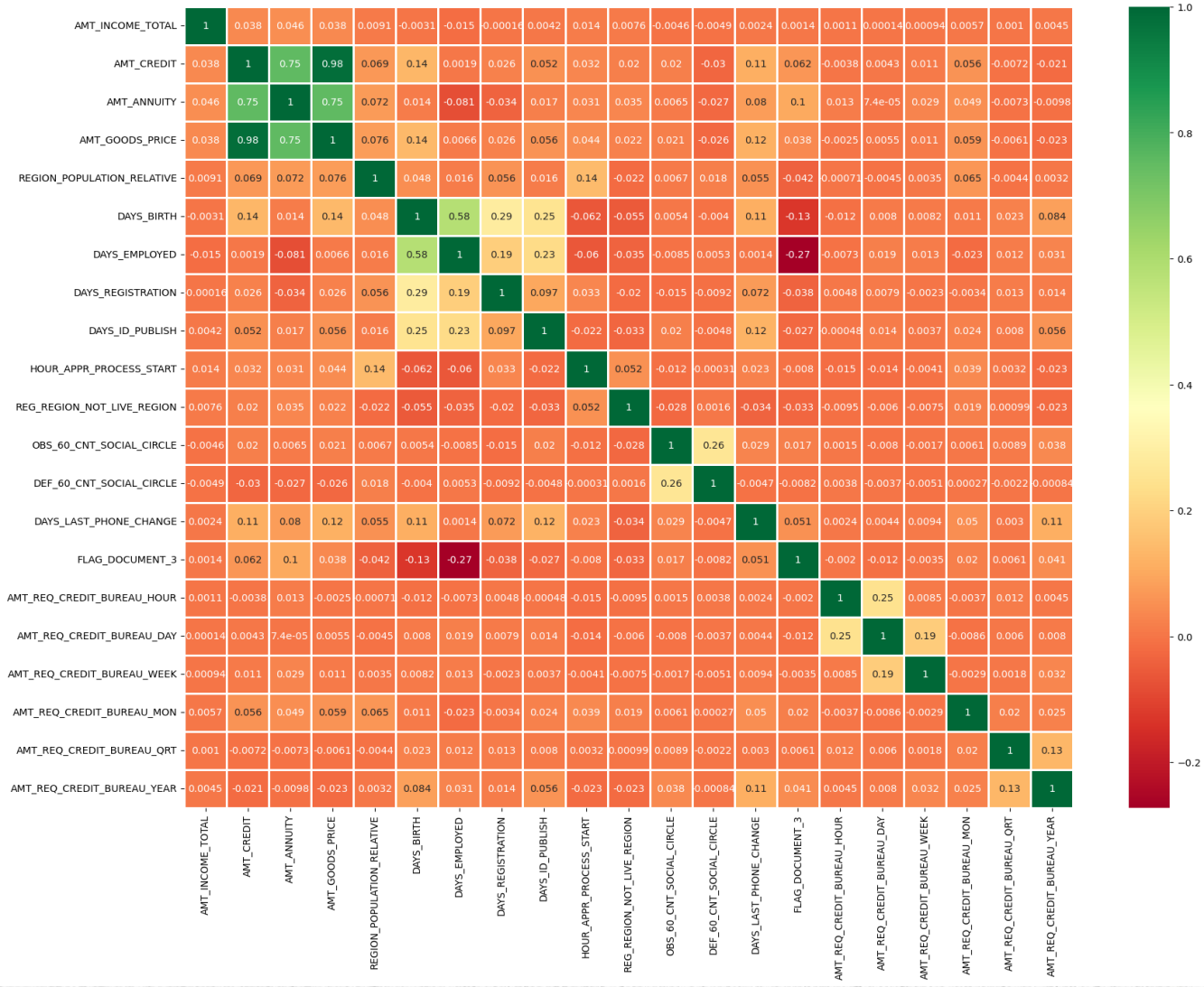
CORRELATION BETWEEN VARIABLES WHEN DEFAULT

	VAR1	VAR2	Correlation
64	AMT_GOODS_PRICE	AMT_CREDIT	0.982783
65	AMT_GOODS_PRICE	AMT_ANNUITY	0.752295
43	AMT_ANNUITY	AMT_CREDIT	0.752195
131	DAYS_EMPLOYED	DAYS_BIRTH	0.582185
152	DAYS_REGISTRATION	DAYS_BIRTH	0.289114
300	FLAG_DOCUMENT_3	DAYS_EMPLOYED	0.272169
263	DEF_60_CNT_SOCIAL_CIRCLE	OBS_60_CNT_SOCIAL_CIRCLE	0.264357
173	DAYS_ID_PUBLISH	DAYS_BIRTH	0.252863
351	AMT_REQ_CREDIT_BUREAU_DAY	AMT_REQ_CREDIT_BUREAU_HOUR	0.247511
174	DAYS_ID_PUBLISH	DAYS_EMPLOYED	0.229090

- Credit amount is highly correlated with good price amount which is same as repaid.

Credit amount is highly correlated with:

- Goods Price Amount
- Loan Annuity
- Total Income


AMT_INCOME_TOTAL 1 0.038 0.046 0.038 0.0091 -0.0031 -0.015 -0.00016 0.0042 0.014 0.0076 -0.0046 -0.0049 0.0024 0.0014 0.0011 0.00014 0.00094 0.0057 0.001 0.0045
AMT_CREDIT 0.038 1 0.75 0.98 0.069 0.14 0.0019 0.026 0.052 0.032 0.02 0.02 -0.03 0.11 0.062 -0.0038 0.0043 0.011 0.056 -0.0072 -0.021
AMT_ANNUITY 0.046 0.75 1 0.75 0.072 0.014 -0.081 -0.034 0.017 0.031 0.035 0.0065 -0.027 0.08 0.1 0.013 7.4e-05 0.029 0.049 -0.0073 -0.0098
AMT_GOODS_PRICE 0.038 0.98 0.75 1 0.076 0.14 0.0066 0.026 0.056 0.044 0.022 0.021 -0.026 0.12 0.038 -0.0025 0.0055 0.011 0.059 -0.0061 -0.023
REGION_POPULATION_RELATIVE 0.0091 0.069 0.072 0.076 1 0.048 0.016 0.056 0.016 0.14 -0.022 0.0067 0.018 0.055 -0.042 -0.00071 -0.0045 0.0035 0.065 -0.0044 0.0032
DAYS_BIRTH -0.0031 0.14 0.014 0.14 0.048 1 0.58 0.29 0.25 -0.062 -0.055 0.0054 -0.004 0.11 -0.13 -0.012 0.008 0.0082 0.011 0.023 0.084
DAYS_EMPLOYED -0.015 0.0019 -0.081 0.0066 0.016 0.58 1 0.19 0.23 -0.06 -0.035 -0.0085 0.0053 0.0014 -0.27 -0.0073 0.019 0.013 -0.023 0.012 0.031
DAYS_REGISTRATION 0.00016 0.026 -0.034 0.026 0.056 0.29 0.19 1 0.097 0.033 -0.02 -0.015 -0.0092 0.072 -0.038 0.0048 0.0079 -0.0023 -0.0034 0.013 0.014
DAYS_ID_PUBLISH 0.0042 0.052 0.017 0.056 0.016 0.25 0.23 0.097 1 -0.022 -0.033 0.02 -0.0048 0.12 -0.027 0.00048 0.014 0.0037 0.024 0.008 0.056
HOUR_APPR_PROCESS_START 0.014 0.032 0.031 0.044 0.14 -0.062 -0.06 0.033 -0.022 1 0.052 -0.012 0.00031 0.023 -0.008 -0.015 -0.014 -0.0041 0.039 0.0032 -0.023
REG_REGION_NOT_LIVE_REGION 0.0076 0.02 0.035 0.022 -0.022 -0.055 -0.035 -0.02 -0.033 0.052 1 -0.028 0.0016 -0.034 -0.033 -0.0095 -0.006 -0.0075 0.019 0.00099 -0.023
OBS_60_CNT_SOCIAL_CIRCLE -0.0046 0.02 0.0065 0.021 0.0067 0.0054 -0.0085 -0.015 0.02 -0.012 -0.028 1 0.26 0.029 0.017 0.0015 -0.008 -0.0017 0.0061 0.0089 0.038
DEF_60_CNT_SOCIAL_CIRCLE -0.0049 -0.03 -0.027 -0.026 0.018 -0.004 0.0053 -0.0092 -0.0048 0.00031 0.0016 0.26 1 -0.0047 -0.0082 0.0038 -0.0037 -0.0051 0.00027 -0.0022 0.00084
DAYS_LAST_PHONE_CHANGE 0.0024 0.11 0.08 0.12 0.055 0.11 0.0014 0.072 0.12 0.023 -0.034 0.029 -0.0047 1 0.051 0.0024 0.0044 0.0094 0.05 0.003 0.11
FLAG_DOCUMENT_3 0.0014 0.062 0.1 0.038 -0.042 -0.13 -0.27 -0.038 -0.027 -0.008 -0.033 0.017 -0.0082 0.051 1 -0.002 -0.012 -0.0035 0.02 0.0061 0.041
AMT_REQ_CREDIT_BUREAU_HOUR 0.0011 -0.0038 0.013 -0.0025 0.00071 -0.012 -0.0073 0.0048 0.00048 -0.015 -0.0095 0.0015 0.0038 0.0024 -0.002 1 0.25 0.0085 -0.0037 0.012 0.0045
AMT_REQ_CREDIT_BUREAU_DAY 0.00014 0.0043 7.4e-05 0.0055 -0.0045 0.008 0.019 0.0079 0.014 -0.014 -0.006 -0.008 -0.0037 0.0044 -0.012 0.25 1 0.19 -0.0086 0.006 0.008
AMT_REQ_CREDIT_BUREAU_WEEK 0.00094 0.011 0.029 0.011 0.0035 0.0082 0.013 -0.0023 0.0037 -0.0041 -0.0075 -0.0017 -0.0051 0.0094 -0.0035 0.0085 0.19 1 -0.0029 0.0018 0.032
AMT_REQ_CREDIT_BUREAU_MON 0.0057 0.056 0.049 0.059 0.065 0.011 -0.023 -0.0034 0.024 0.039 0.019 0.0061 0.00027 0.05 0.02 -0.0037 -0.0086 -0.0029 1 0.02 0.025
AMT_REQ_CREDIT_BUREAU_QRT 0.001 -0.0072 -0.0073 -0.0061 -0.0044 0.023 0.012 0.013 0.008 0.0032 0.00099 0.0089 -0.0022 0.003 0.0061 0.012 0.006 0.0018 0.02 1 0.13
AMT_REQ_CREDIT_BUREAU_YEAR 0.0045 -0.021 -0.0098 -0.023 0.0032 0.084 0.031 0.014 0.056 -0.023 -0.023 0.038 0.00084 0.11 0.041 0.0045 0.008 0.032 0.025 0.13 1
AMT_INCOME_TOTAL
AMT_CREDIT
AMT_ANNUITY
AMT_GOODS_PRICE
REGION_POPULATION_RELATIVE
DAYS_BIRTH
DAYS_EMPLOYED
DAYS_REGISTRATION
DAYS_ID_PUBLISH
HOUR_APPR_PROCESS_START
REG_REGION_NOT_LIVE_REGION
OBS_60_CNT_SOCIAL_CIRCLE
DEF_60_CNT_SOCIAL_CIRCLE
DAYS_LAST_PHONE_CHANGE
FLAG_DOCUMENT_3
AMT_REQ_CREDIT_BUREAU_HOUR
AMT_REQ_CREDIT_BUREAU_DAY
AMT_REQ_CREDIT_BUREAU_WEEK
AMT_REQ_CREDIT_BUREAU_MON
AMT_REQ_CREDIT_BUREAU_QRT
AMT_REQ_CREDIT_BUREAU_YEAR
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

TASK-6 FINAL SUMMARY CONCLUSION

After analysing the datasets, there are few attributes of a client with which the bank would be able to identify if they will repay the loan or not.

DECISIVE FACTOR WHETHER AN APPLICANT WILL BE REPAYER

1. NAME_EDUCATION_TYPE: Academic degree has less defaults.
2. NAME_INCOME_TYPE: Student and Businessmen have no defaults.
3. REGION_RATING_CLIENT: RATING 1 is safer.
4. ORGANIZATION_TYPE: Clients with Trade Type 4 and 5 and Industry type 8 have defaulted less than 3%.
5. DAYS_BIRTH: People above age of 50 have low probability of defaulting.
6. DAYS_EMPLOYED: Clients with 40+ year experience having less than 1% default rate.
7. AMT_INCOME_TOTAL:Applicant with Income more than 700,000 are less likely to default.
8. NAME_CASH_LOAN_PURPOSE: Loans bought for Hobby, Buying garage are being repaid mostly.
9. CNT_CHILDREN: People with zero to two children tend to repay the loans.

DECISIVE FACTOR WHETHER AN APPLICANT WILL BE DEFAULTER

1. CODE_GENDER: Men are at relatively higher default rate.
2. NAME_FAMILY_STATUS : People who have civil marriage or who are single default a lot.
3. NAME_EDUCATION_TYPE: People with Lower Secondary & Secondary education.
4. NAME_INCOME_TYPE: Clients who are either at Maternity leave OR Unemployed default a lot.
5. REGION_RATING_CLIENT: People who live in Rating 3 has highest defaults.
6. OCCUPATION_TYPE: Avoid Low-skill Laborers, Drivers and Waiters/barmen staff, Security staff, Laborers and Cooking staff as their default rate is huge.
7. ORGANIZATION_TYPE: Organizations with highest percent of loans not repaid are Transport: type 3 (16%), Industry: type 13 (13.5%), Industry: type 8 (12.5%) and Restaurant (less than 12%). Self-employed people have relative high defaulting rate, and thus should be avoided to be approved for loan or provide loan with higher interest rate to mitigate the risk of defaulting.
8. DAYS_BIRTH: Avoid young people who are in age group of 20-40 as they have higher probability of defaulting.
9. DAYS_EMPLOYED: People who have less than 5 years of employment have high default rate.
10. CNT_CHILDREN & CNT_FAM_MEMBERS: Client who have children equal to or more than 9 default 100% and hence their applications are to be rejected.
11. AMT_GOODS_PRICE: When the credit amount goes beyond 3lakhs, there is an increase in defaulters.

FACTORS THAT LOAN CAN BE GIVEN ON CONDITION OF HIGH INTEREST RATE TO MITIGATE ANY DEFAULT RISK LEADING TO BUSINESS LOSS:

1. NAME_HOUSING_TYPE: High number of loan applications are from the category of people who live in Rented apartments & living with parents and hence offering the loan would mitigate the loss if any of those default.
2. AMT_CREDIT: People who get loan for 3-6 Lakhs tend to default more than others and hence having higher interest specifically for this credit range would be ideal.
3. AMT_INCOME: Since 90% of the applications have Income total less than 3Lakhs and they have high probability of defaulting, they could be offered loan with higher interest compared to other income category.
4. CNT_CHILDREN & CNT_FAM_MEMBERS: Clients who have 4 to 8 children has a very high default rate and hence higher interest should be imposed on their loans.
5. NAME_CASH_LOAN_PURPOSE: Loan taken for the purpose of Repairs seems to have highest default rate. A very high number applications have been rejected by bank or refused by client in previous applications as well which has purpose as repair or other. This shows that purpose repair is taken as high risk by bank and either they are rejected, or bank offers very high loan interest rate which is not feasible by the clients, thus they refuse the loan. The same approach could be followed in future as well.

SUGGESTIONS

- 90% of the previously cancelled client have actually repaid the loan. Record the reason for cancellation which might help the bank to determine and negotiate terms with these repaying customers in future for increase business opportunity.
- 88% of the clients who were refused by bank for loan earlier have now turned into a repaying client. Hence documenting the reason for rejection could mitigate the business loss and these clients could be contacted for further loans.

Impact Of Car Features

Description

- The automotive industry has been rapidly evolving over the past few decades with a growing focus on fuel efficiency, environmental sustainability, and technology innovation. With increasing competition in the market, manufacturers want to change the consumer landscape.
- This project objective is to analyze the car features that affect the manufacturers' car prices. As a Data Analyst, the client has asked how can a car manufacturer optimize pricing and product development decisions to maximize profitability while meeting consumer demand.
- Dataset contains information on various car models and their applications. The dataset is available on Kaggle by Cooper College, New York City.
- Total number of observations:- 11,813
- File type:- CSV

Tech Stack Used:- MS-Excel

APPROACH

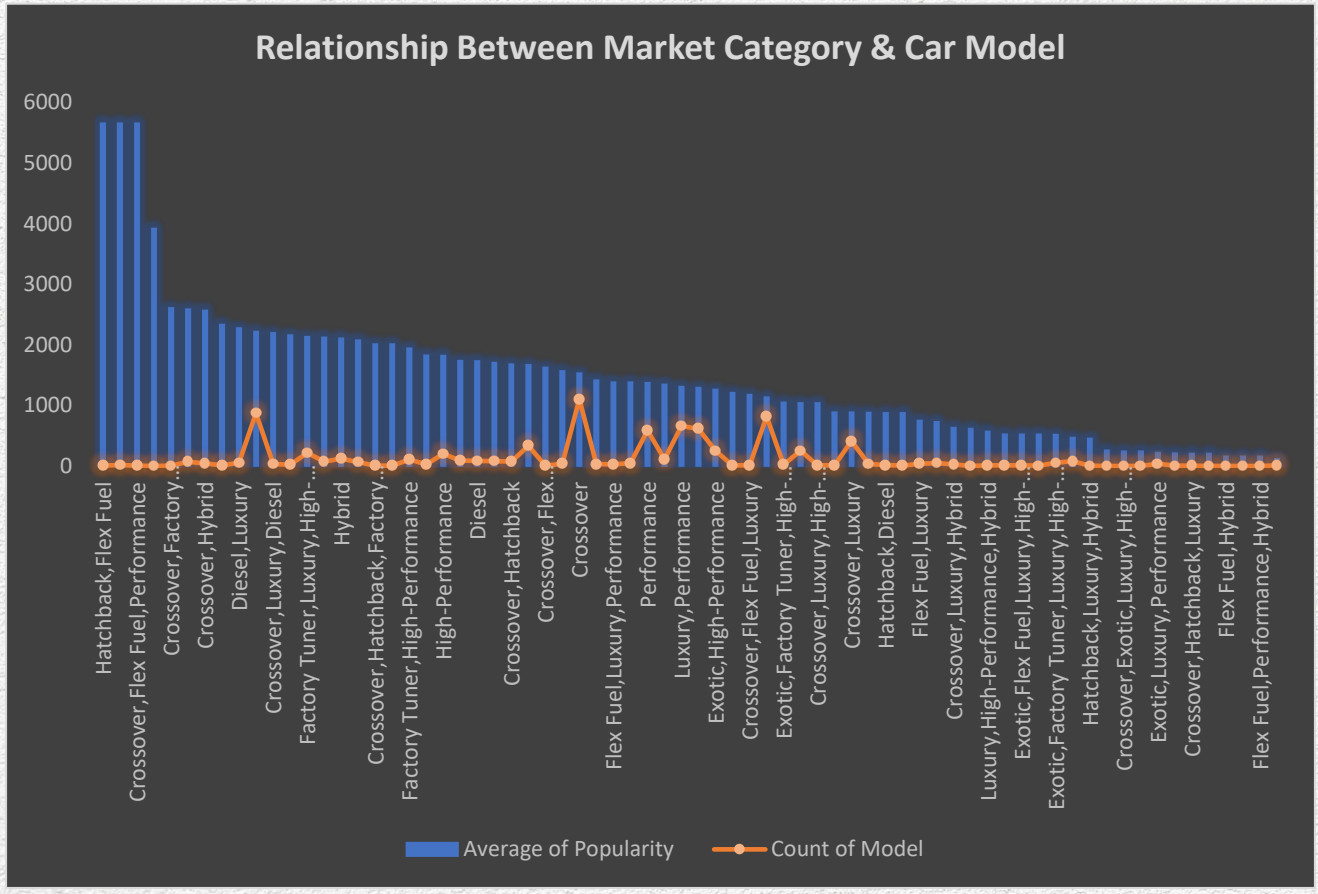
For analysis I used descriptive statistics, regression analysis, and visualization to visualize the charts and graphs.

The reason behind using descriptive analysis was to find out the mean of the car's features variable and then visualize them in the form of charts and graphs.

Regression Analysis told the dependency of independent variables to dependent variables i.e., the specific features of the car impact the car's price.

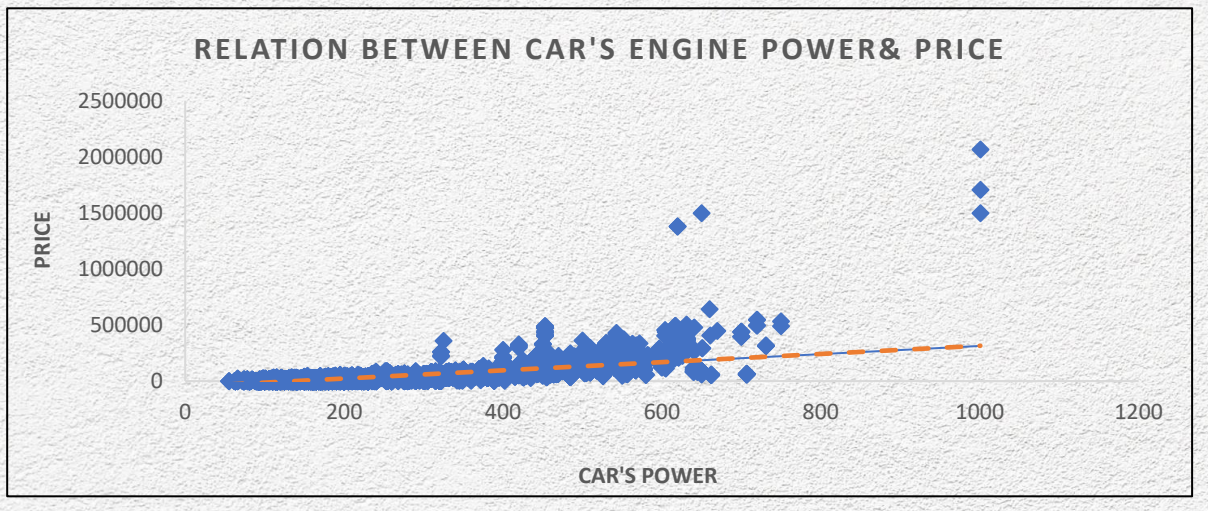
While creating the charts it was difficult to summarize the values of variables in sum or average.

Insight Required: How does the popularity of a car model vary across different market categories?



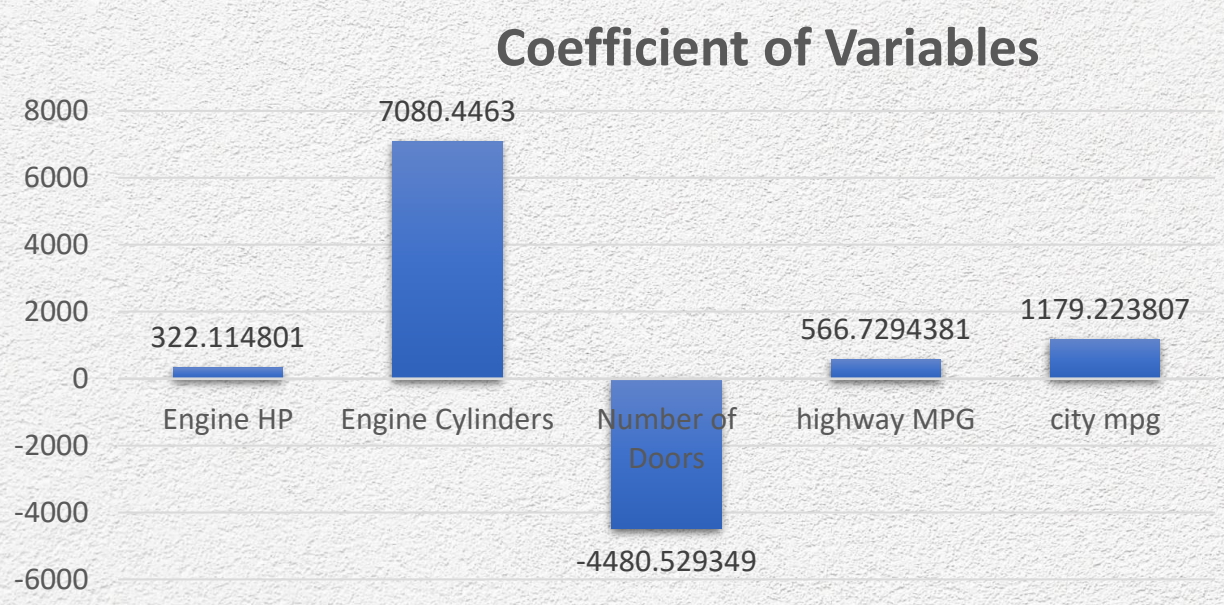
From the above line chart, we can see that Hatchback & Flex Fuel are the highest populated and have the highest number of cars.

Insight Required: What is the relationship between a car's engine power and its price?



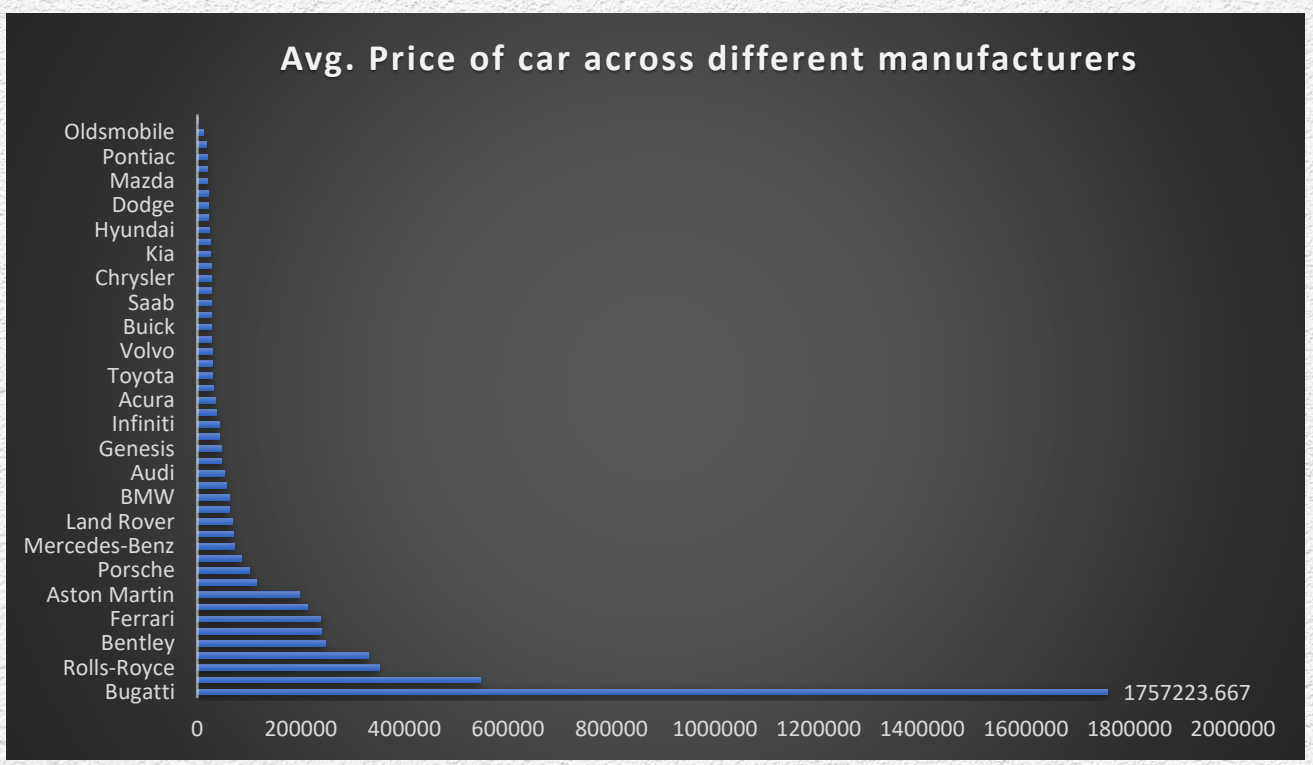
From the above scatter plot, we can observe that cars that have high engine power have higher prices. Thus, the car's power increases as the car's price will also increase.

Insight Required: Which car features are most important in determining a car's price?



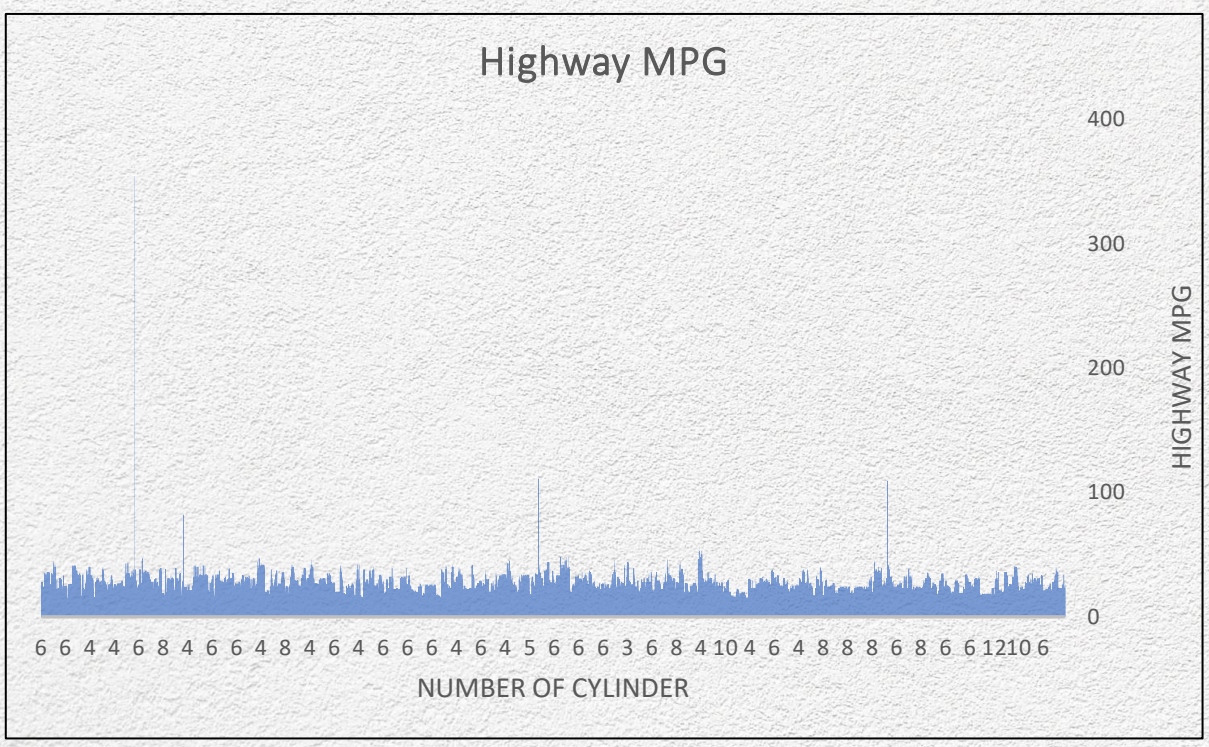
From the above bar chart, we can see that Engine Cylinders have the strongest relationship with MSRP.

Insight Required: How does the average price of a car vary across different manufacturers?



From the above bar chart, we can say that Bugatti has the highest average price.

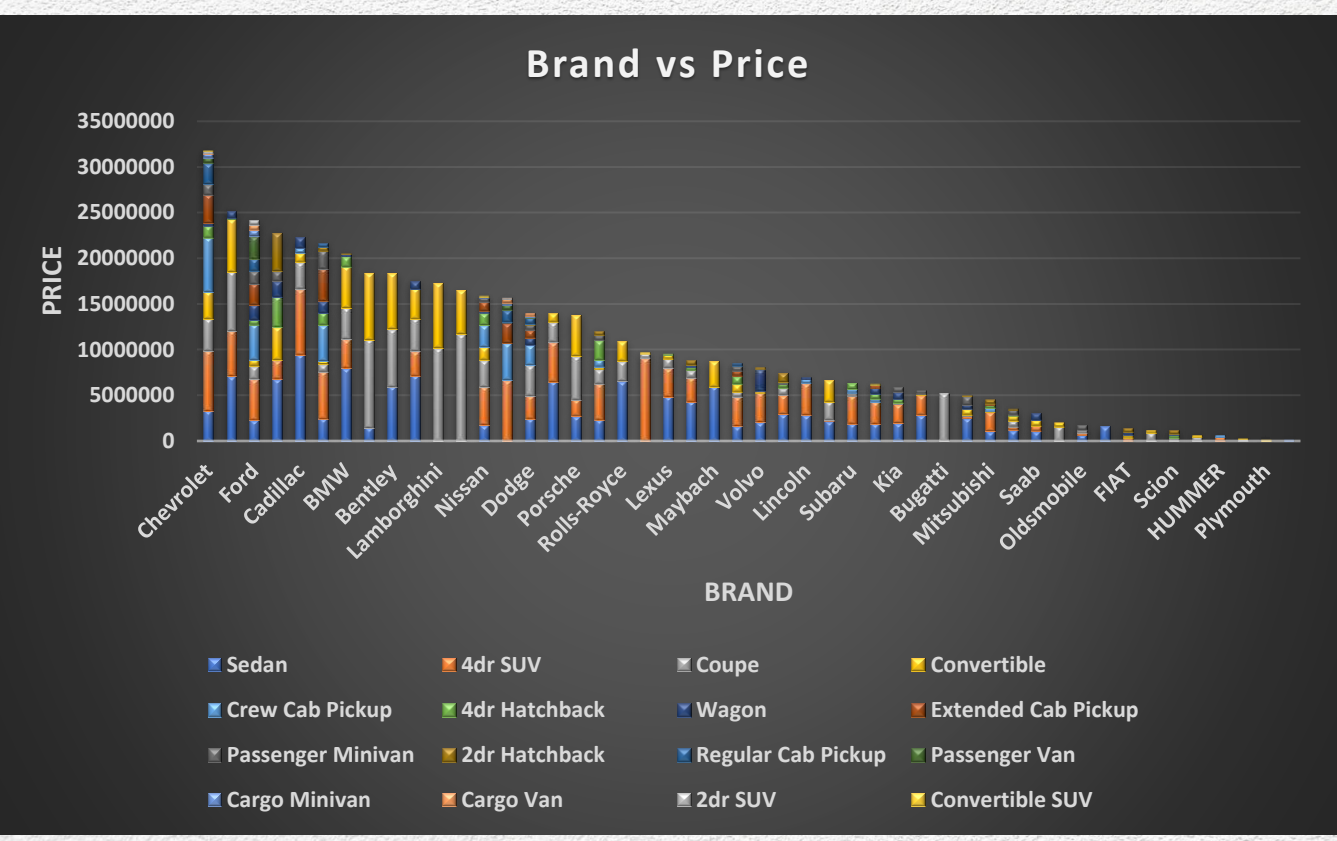
Insight Required: What is the relationship between fuel efficiency and the number of cylinders in a car's engine?



From the above chart, we can see that cylinders having 4 is giving maximum fuel efficiency.
And Fuel efficiency is inversely proportional to the no. of cylinders.

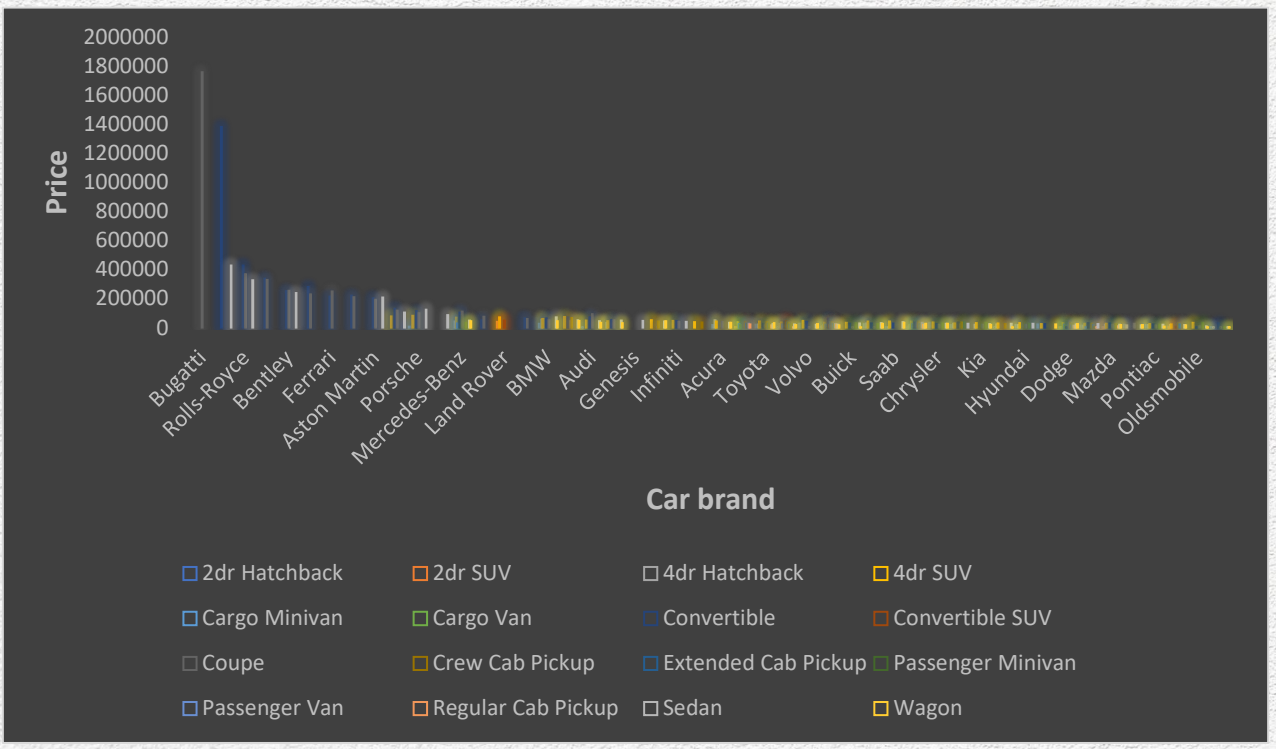
Building the Dashboard

Task 1: How does the distribution of car prices vary by brand and body style?



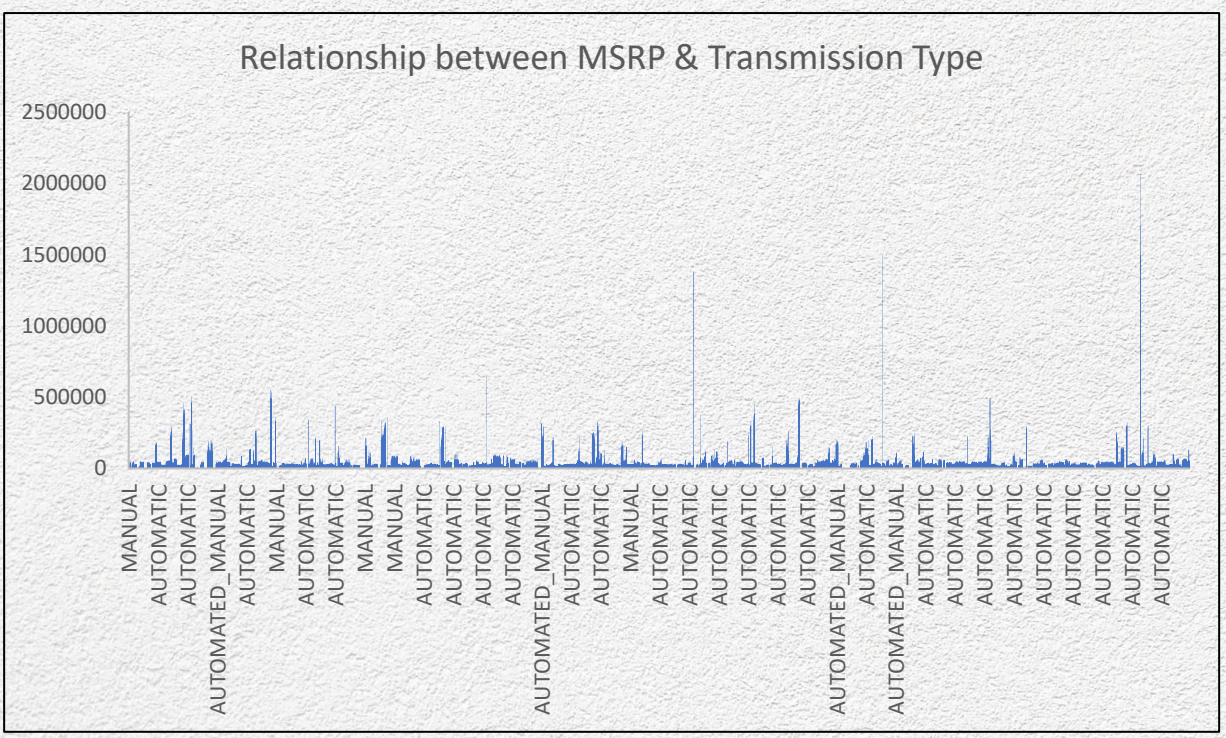
From the above-stacked column, Chevrolet and Mercedes-Benz have the highest contribution to the car's price.

Task 2: Which car brands have the highest and lowest average MSRPs, and how does this vary by body style?



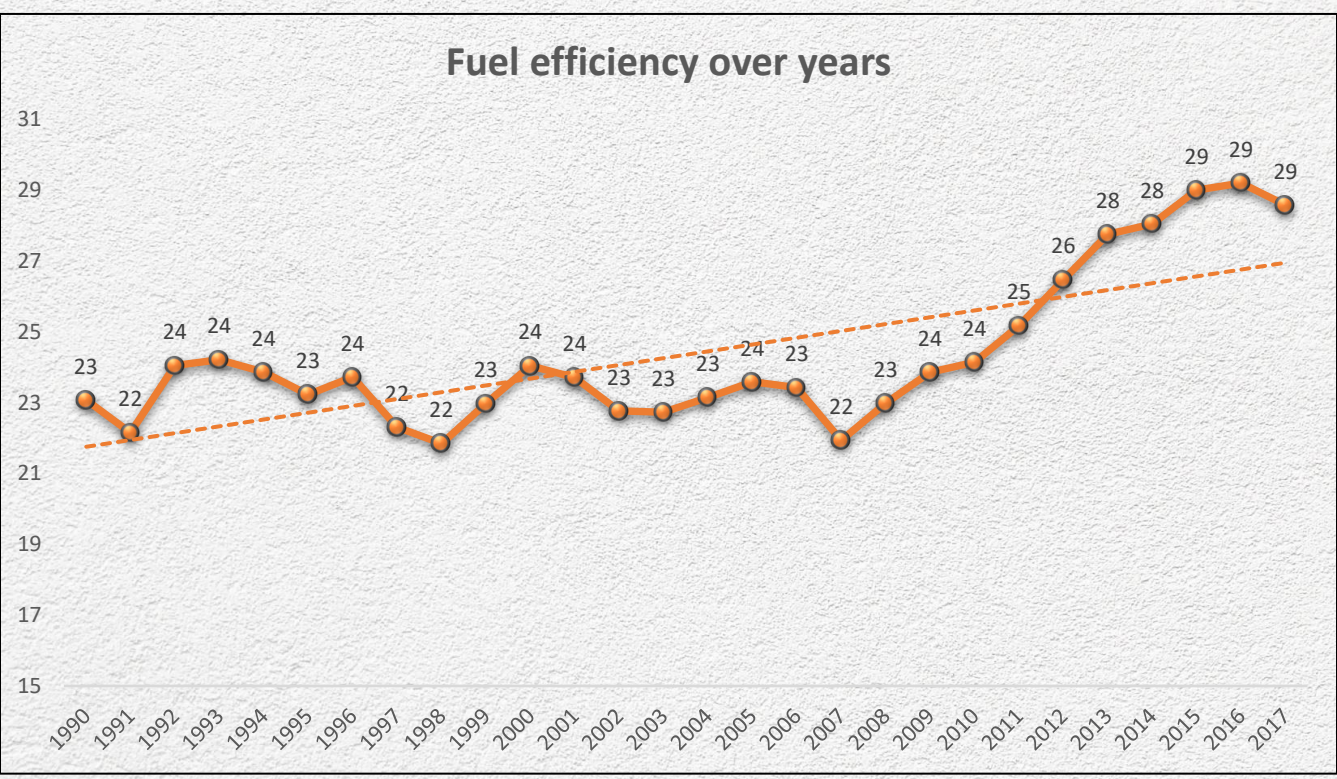
From the above-clustered chart, we can see that the couple style Bugatti and the Convertible style of Maybach have the highest average car prices.

Task 3: How do the different features such as transmission type affect the MSRP, and how does this vary by body style?



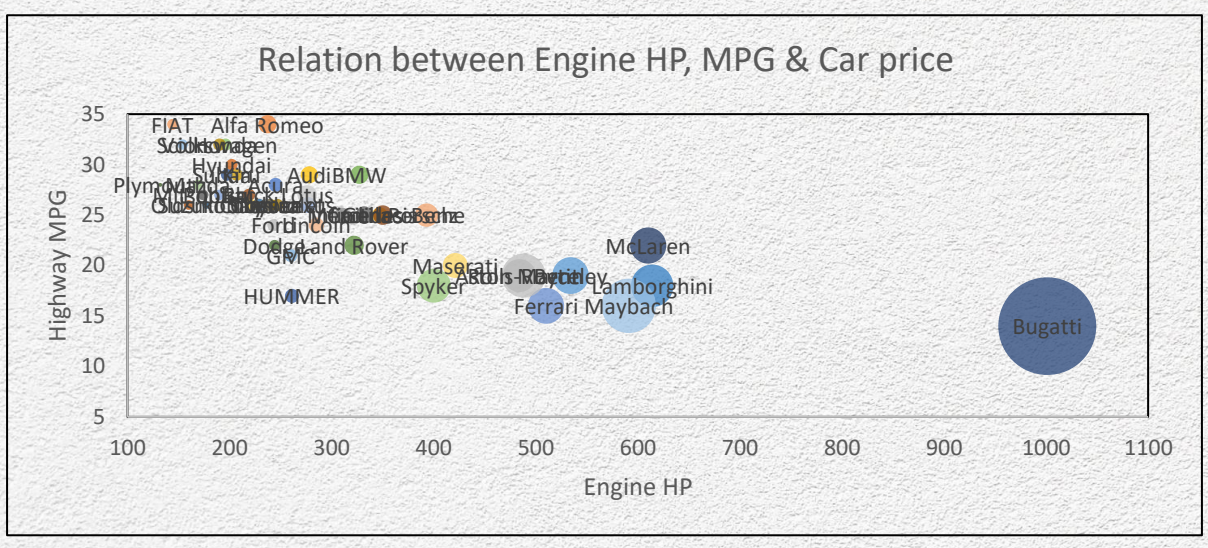
From the above chart, we can see that the automated convertible and automated_manual couple are highly contributing in MSRP.

Task 4: How does the fuel efficiency of cars vary across different body styles and model years?

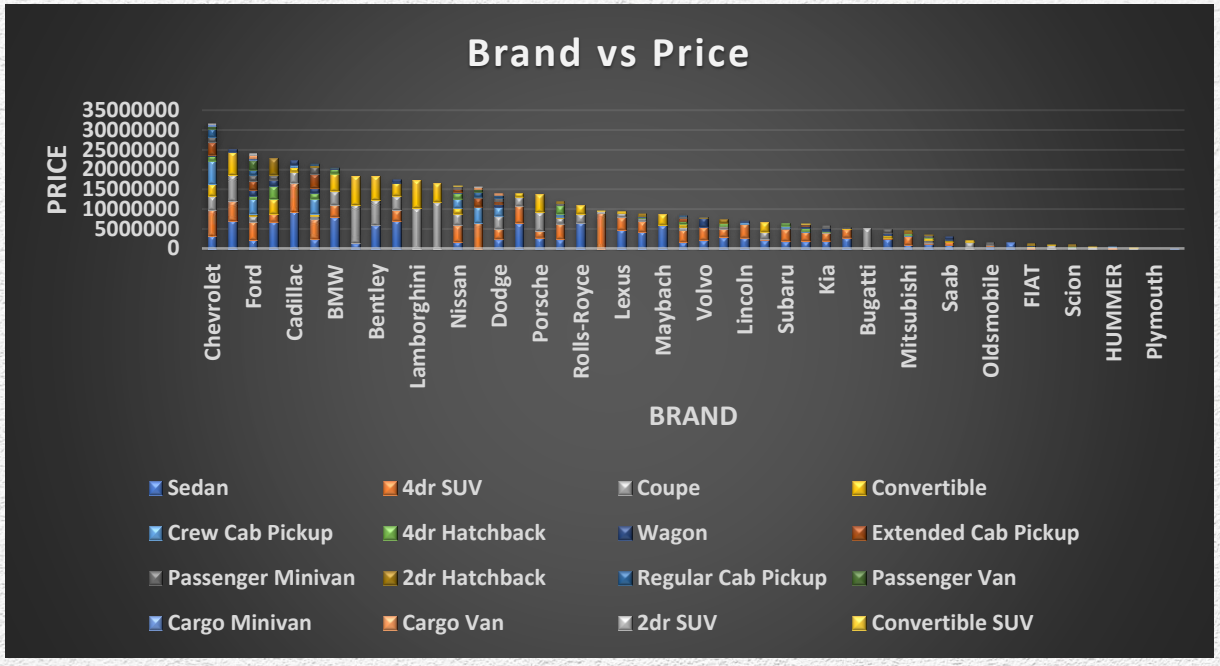


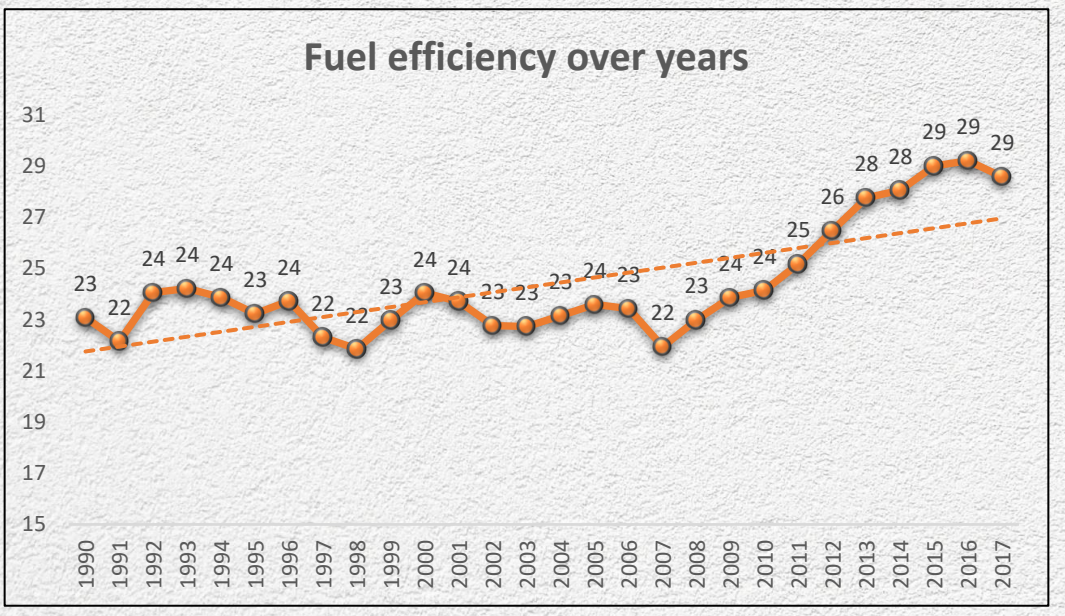
From the above line chart, we can see that as time increases fuel efficiency (Highway MPG)also increased or improved.

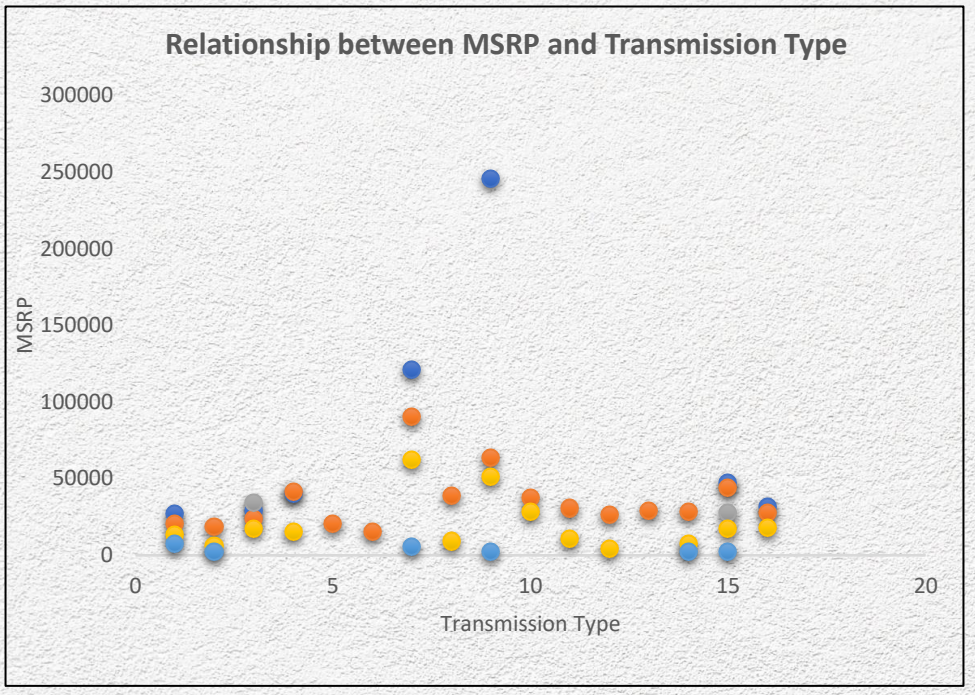
Task 5: How does the car's horsepower, MPG, and price vary across different Brands?

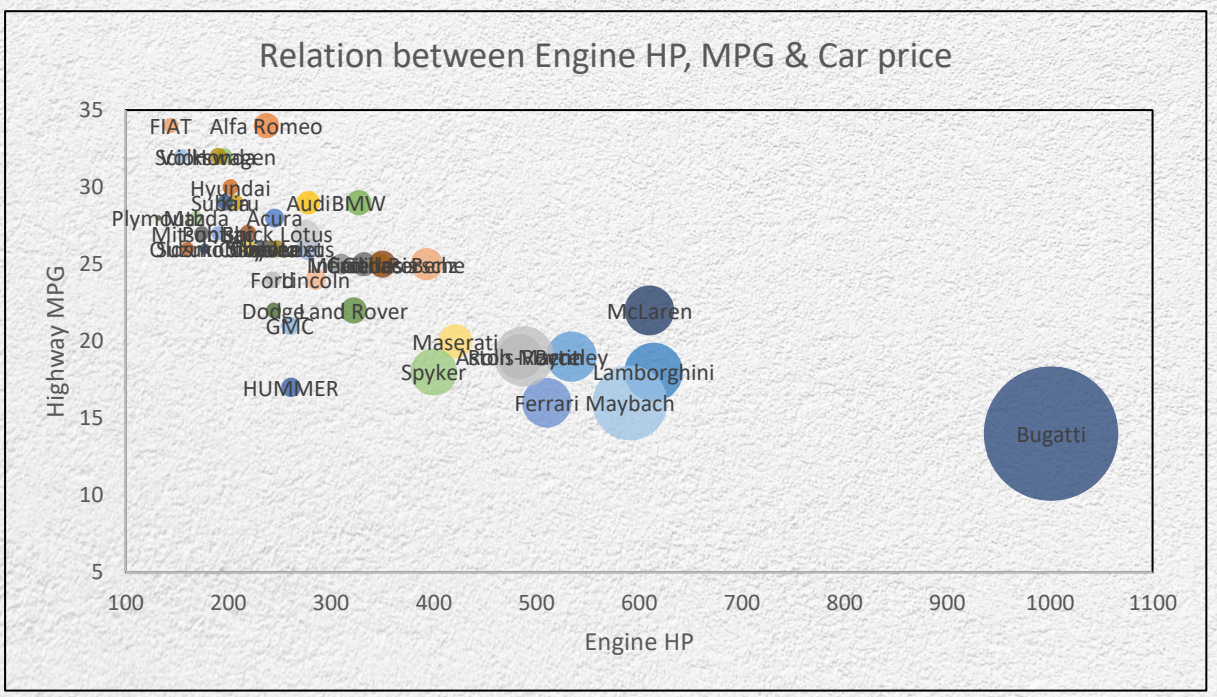


From the above Bubble chart, we can see that if engine hp increase, highway mpg will decrease and the price will also increase.


Brand vs Price
PRICE
35000000
30000000
25000000
20000000
15000000
10000000
5000000
0
Chevrolet
Ford
Cadillac
BMW
Bentley
Lamborghini
Nissan
Dodge
Porsche
Rolls-Royce
Lexus
Maybach
Volvo
Lincoln
Subaru
Kia
Bugatti
Mitsubishi
Saab
Oldsmobile
FIAT
Scion
HUMMER
Plymouth
BRAND
Sedan
4dr SUV
Coupe
Convertible
Crew Cab Pickup
4dr Hatchback
Wagon
Extended Cab Pickup
Passenger Minivan
2dr Hatchback
Regular Cab Pickup
Passenger Van
Cargo Minivan
Cargo Van
2dr SUV
Convertible SUV


Fuel efficiency over years
31
29
27
25
23
21
19
17
15
23 22 24 24 24 23 24 22 22 23 24 24 23 23 23 24 23 22 23 24 24 25 26 28 28 29 29 29
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017


Relationship between MSRP and Transmission Type
300000
250000
200000
150000
100000
50000
0
MSRP
0
5
10
15
20
Transmission Type


Relation between Engine HP, MPG & Car price
35
30
25
20
15
10
5
Highway MPG
100
200
300
400
500
600
700
800
900
1000
1100
Engine HP
FIAT
Alfa Romeo
Hyundai
Audi
BMW
Acura
Lotus
Ford
Lincoln
Dodge
Land Rover
GMC
McLaren
Maserati
Spyker
Lamborghini
HUMMER
Ferrari
Maybach
Bugatti

Result

- Coupe body style contributing maximum to the car's MSRP.
- In transmission type automated_manual creates high impact because in a single car having both automated and manual gear systems will be more beneficial rather than a single gear system.
- Overall, fuel efficiency and Coupe body style features highly impact the car's price and profitability.
- Companies need to produce high or at least good fuel efficiency of cars by which the majority of the class can afford a car.

- File link:-Click To See
- Dashboard sheet - Link To Dashboard

ABC Call Volume Trend

Description

This project is about how company uses its analytical skills to target audiences from many types of media platform to convert them as their customers at low cost.
I am provided with **Customer Experience (CX) Inbound calling team for 23 days (Call_Volume_Trend_Analysis)** data sets, tables from which I must derive certain insights out of it and answer the questions. so it will be easy for me to handle it using **Excel** and provide a detailed report

2) Approach

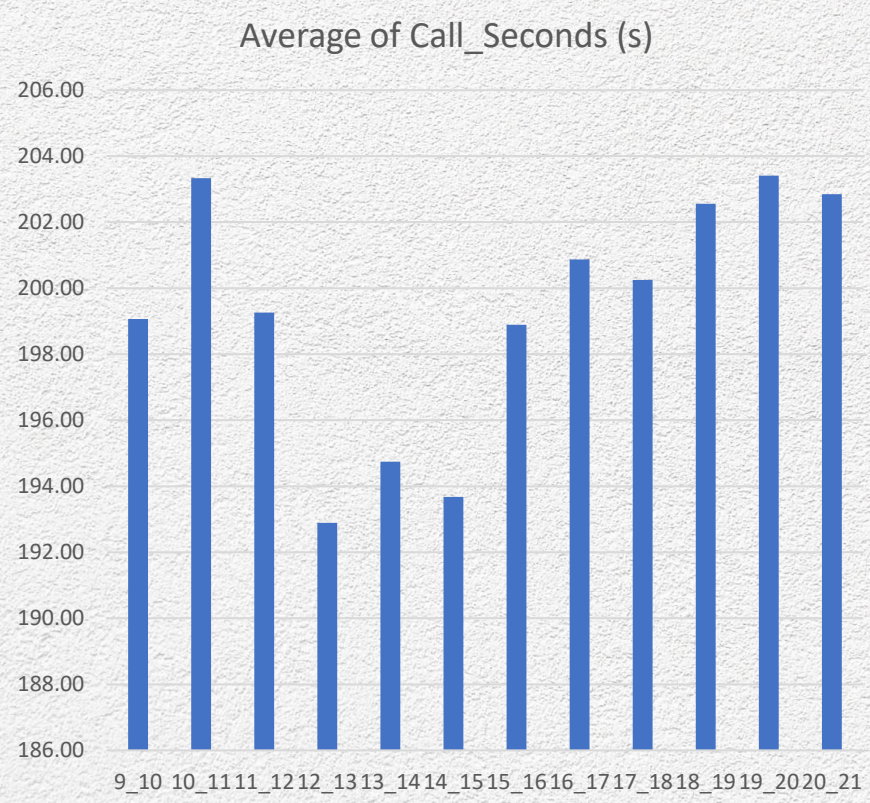
1. I revised 2-3 times the Description of Final Project-4 (ABC Call Volume Trend Analysis).
2. Collected the **Call_Volume_Trend_Analysis** Dataset.
3. Inserted/Loaded it into Excel.
4. Finally analyzed the dataset and attached results for the given questions.

3) Tech-Stack Used

- I have used **Excel** software.
- Excel is a tool **for organizing and performing calculations** on data.
- It can analyze data, calculate statistics, generate **pivot table**, and represent data as **charts or graphs.**
- I have used **Excel 2021 version** to complete this project.

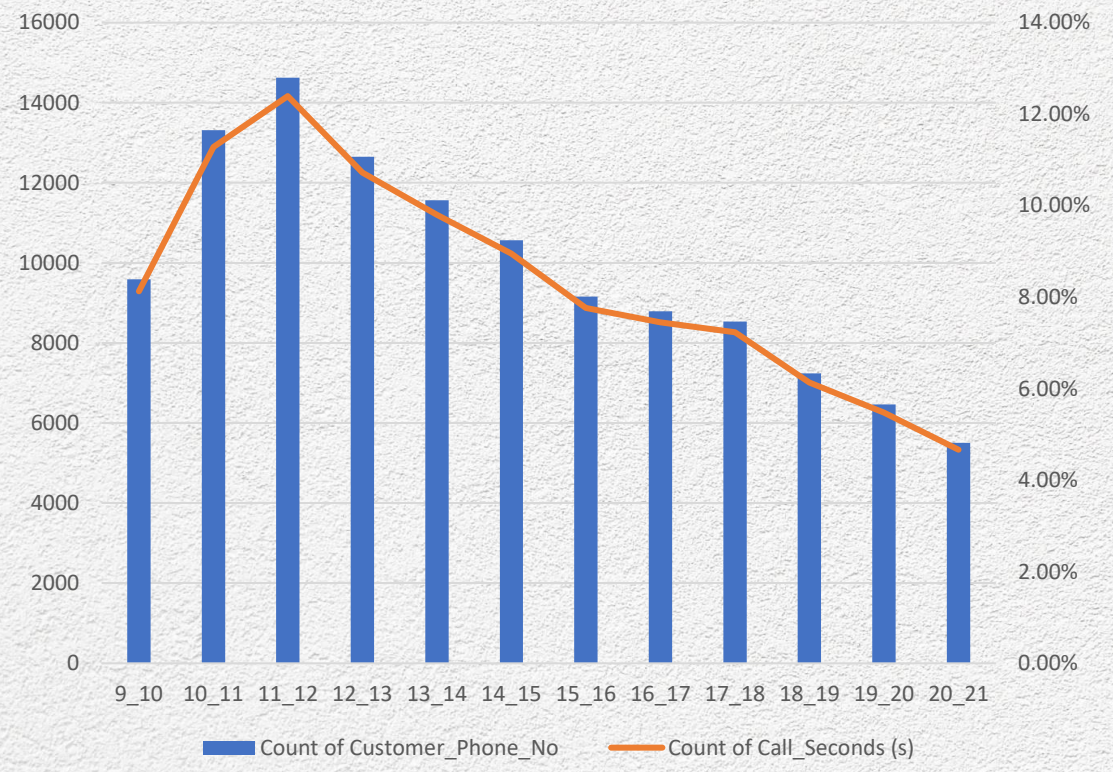
4) Insights

1) Calculate the average call time duration for all incoming calls received by agents (in each Time Bucket).



- I have put Time_Bucket in Row Section and computed average of Call_Seconds in Value Section and plotted a Bar Graph.
- Agents answer calls for an average of 198.6 seconds.
- Average call time duration is highest between 7pm-8pm with value of 203.41 and 10 am to 11 am with value of 203.33.
- The average call time duration is least between 12am and 1 pm with value of 192.89.

2) Show the total volume/ number of calls coming in via charts/ graphs [Number of calls v/s Time]. You can select time in a bucket form (i.e. 1-2, 2-3, ...)



I have put Time_Bucket in Row Section and computed count of Customer_Phone_No and count of Call_Seconds in Value Section and plotted Clustered Cloumn chart where

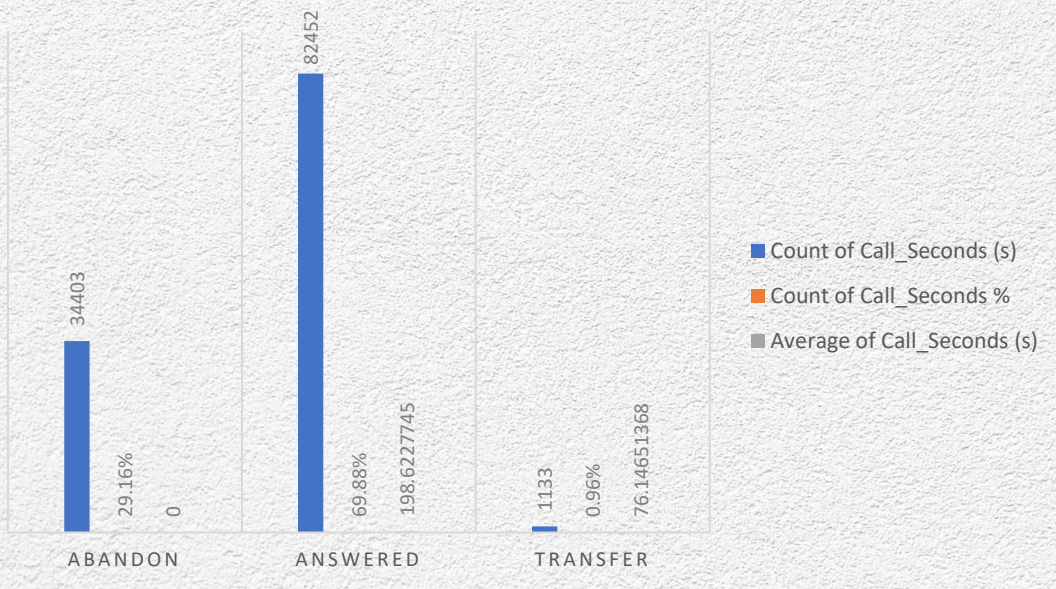
Bar Graph represents Count of Customer_Phone_No

Line Graph represents Count of Call_Seconds

Customers call the most in between 11am to 12am.

Customers call the least in between 8pm to 9pm.

3) As you can see current abandon rate is approximately 30%. Propose a manpower plan required during each time bucket [between 9am to 9pm] to reduce the abandon rate to 10%. (i.e. You have to calculate minimum number of agents required in each time bucket so that at least 90 calls should be answered out of 100.)



Assumption				
	Time (in Hrs)			Time (in Mins)
The ABC Company's Total Working Hours	9		Morning Break	15
Break	1.5		Lunch	60
IT DownTime	0.5		Afternoon Break	15
Meetings/Discussions	1			
Agent's actual Working Hours	5			

Date	Sum of Call_Seconds (s)	Sum of Call_Seconds (s)2
01-Jan	676664	187.96

- Total agents working is calculated by average calls on single day divided by total time spend by an agent in a single day.

  Total Agent = 187.96/5 = 37.59 or 38

- If Agents are working for 5 hrs a day and 60% of calls are answered.
- If the ABC Company want 90% of calls to get connected, using unitary method we can determine how many agents are needed.

  Agents Required = 30%-20% = 10%

Additional Agents Required – 20%/30%*38 = 25.33 or 26 Approx

Time Bucket	Count of Call Sec	Count of Call Sec	Manpower Required
9_10	8.13%	0.085	10 (approx.)
10_11	11.28%	0.116	12 (approx.)
11_12	12.40%	0.127	13 (approx.)
12_13	10.72%	0.116	12 (approx.)
13_14	9.80%	0.106	11 (approx.)
14_15	8.95%	0.095	10 (approx.)
15_16	7.76%	0.084	9 (approx.)
16_17	7.45%	0.074	8 (approx.)
17_18	7.23%	0.074	8 (approx.)
18_19	6.13%	0.063	7 (approx.)
19_20	5.48%	0.053	6 (approx.)
20_21	4.67%	0.053	6 (approx.)
Total	56		

I have created a pivot table with Call_Status in Row Section and count of Call_Seconds and in percentage as well and Average of Call_Seconds. Then plotted a Bar Graph.

To calculate I have created a pivot table with Date & Time in Row Section and Sum of Call_Seconds in value Section. As the values are in Seconds I have divided them with 3600 to convert them to Hours. 29.16% of the calls are abandoned, 69.88% of the calls are answered and 0.96% of the call are transferred.

To achieve a 90% call connection rate (instead of the current 60%), we calculate the number of additional agents needed. Applying the unitary method, we find that approximately 56 agents would be required.

56 Agents are needed to answer 90% of calls per day.

4) Let's say customers also call this ABC insurance company in night but didn't get answer as there are no agents to answer, this creates a bad customer experience for this Insurance company. Suppose every 100 calls that customer made during 9 Am to 9 Pm, customer also made 30 calls in night between interval [9 Pm to 9 Am] and distribution of those 30 calls are as follows. Now propose a manpower plan required during each time bucket in a day. Maximum Abandon rate assumption would be same 10%

Distribution of 30 calls coming in night for every 100 calls coming in between 9am - 9pm (i.e. 12 hrs slot)											
9pm- 10pm	10pm - 11pm	11pm- 12am	12am- 1am	1am - 2am	2am - 3am	3am - 4am	4am - 5am	5am - 6am	6am - 7am	7am - 8am	8am - 9am
3	3	2	2	1	1	1	1	3	4	4	5

I have created a pivot table with Date_&_Time in Row Section, count of Customer_Phone_No and Call_Status in Column Section.
Average calls in DayTime is calculated.

Using the above result Average calls in NightTime is calculated (i.e 30% of Average call in DayTime)

Using Average calls in NigthTime I have calculated Additional Hour required (i.e Average calls in NightTime * Answered Average of Call_Second * 0.9 / 3600)

Using above result I have calculated Additional Agents Required. (i.e Additional Hour required /60% of 7.5hrs)

Average call in DayTime (9am to 9pm)	5130
Average call in NightTime (9pm - 9am)	1539
Additional Hour required	76.4
Additional Agents	17

Time	Call Count	Time Distribution	Agents Required
9pm - 10pm	3	0.10	2
10pm - 11pm	3	0.10	2
11pm - 12pm	2	0.07	1
12pm - 1am	2	0.07	1
1am - 2am	1	0.03	1
2am - 3am	1	0.03	1
3am - 4am	1	0.03	1
4am - 5am	1	0.03	1
5am - 6am	3	0.10	2
6am - 7am	4	0.13	2
7am - 8am	4	0.13	2
8am - 9am	5	0.17	3
Total	30	1	19

Time Distribution is calculated by dividing each Calls Distribution by Total Calls.

The number of agents required for each Time_Bucket is calculated by Additional agents required * Time Distribution.

For the night shift, the ABC Company should hire 17 agents.

Between 1 am to 5 am, customers call the least. As a result, the ABC Company can use a few agents to answer calls at that time.

In order to answer the most calls, the Head can switch some Agent's shifts from 5 am to 2 pm and 2 pm to 11 pm as most of the customers call in these time.

Link To Excel Sheet - Click To See

Conclusion

In conclusion, I would like to tell that after doing a thorough analysis we were able to derive the insights
from the data and was able to plot various graphs using that data. The data which once looked useless
gave some very useful insights.

Thank You

Appendix

Data Analytics Process ---------- Click Here

Instagram User Analytics ---------- Click Here

Operations & Metric Analytics ---------- Click Here

Hiring Process Analytics ---------- Click Here

IMDB Movie Analysis ---------- Click Here

Bank Loan Case Study ---------- Click Here

Impact Of Car Features ---------- Click Here

ABC Call Volume Trend ---------- Click Here