

# Professional Background

I am Manan Yadav, currently in my final year in my 4-year BE in Electrical Engineering from Sardar Vallabhbhai Patel Institute of Technology, Vasad (CGPA 8.11/10). During this course, I got exposure to the subjects like Electrical Machines, Electric Hybrid vehicles, Switchgear and Protection and Power Systems, etc. I have done my Diploma in Electrical engineering.

I have also worked on a Project ‘Distance measurement using ultrasonic sensor’.

As I am a fresher it would be great to experience the real challenges of the corporate world and understand how things work. Being a fresher, I think I am very flexible and adaptive to learn new things. I have theoretical knowledge. But I am waiting to use my theoretical knowledge in a practical way. And I believe by putting significant efforts I will learn.

# Contents

**Table of Contents**

[Professional Background 1](#_bookmark0)

[Contents 2](#_bookmark1)

[Module 1: Data Analytics Process 4](#_bookmark2)

[Overview 4](#_bookmark3)

[Data Analytics Process 4](#_bookmark4)

1. [Ask 4](#_bookmark5)
2. [Prepare: 4](#_bookmark6)
3. [Process 5](#_bookmark7)
4. [Analyze: 5](#_bookmark8)
5. [Share 5](#_bookmark9)
6. [Act 5](#_bookmark10)

[Module 2: Instagram User Analytics 6](#_bookmark11)

[Project Description 6](#_bookmark12)

[Approach: 6](#_bookmark13)

[Tech-Stack Used 6](#_bookmark14)

[Insights 6](#_bookmark15)

[Result 6](#_bookmark16)

[Module 3: Operation and Metric Analytics 10](#_bookmark17)

[Description: 10](#_bookmark18)

[Approach: 10](#_bookmark19)

[Tech-Stack used 10](#_bookmark20)

[Insights 10](#_bookmark21)

[Results 10](#_bookmark22)

[Module 4: Hiring Process Analytics 19](#_bookmark23)

[Project Description 19](#_bookmark24)

[Approach: 19](#_bookmark25)

[Tech-Stack Used 19](#_bookmark26)

[Insights 19](#_bookmark27)

[Result 19](#_bookmark28)

[Module 5: IMDB Movie Analysis 22](#_bookmark29)

[Project Description 22](#_bookmark30)

[Approach: 22](#_bookmark31)

[Tech-Stack Used 22](#_bookmark32)

[Insights 22](#_bookmark33)

[Results 22](#_bookmark34)

[Module 6: Bank Loan Case Study 30](#_bookmark35)

[Description: 30](#_bookmark36)

[Approach: 30](#_bookmark37)

[Tech used 31](#_bookmark38)

[Insights 31](#_bookmark39)

[Results 32](#_bookmark40)

[Conclusion: 82](#_bookmark41)

[Challenges 82](#_bookmark42)

[Module 7: Impact of Car Features Analysis 83](#_bookmark43)

[Project Description 83](#_bookmark44)

[Approach: 83](#_bookmark45)

[Tech-Stack Used 83](#_bookmark46)

[Insights 84](#_bookmark47)

[Results 92](#_bookmark48)

[Module 8: ABC Call Volume Trend Analysis 93](#_bookmark50)

[Project Description 93](#_bookmark51)

[Approach: 93](#_bookmark52)

[Tech-Stack Used 93](#_bookmark53)

[Insights & results 94](#_bookmark54)

[Challenges ……………………………………...98](#_bookmark55)

**Overview:**

# Module 1: Data Analytics Process

We have to buy the best possible house from all the options the broker shows us. I have all the data about the things we need in a house which meets all of our requirements, So I need to make sure I analyze all the data I have about our needs and the budget we have and choose the house which meets both our needs and our budget and communicate that to the broker. Thus using data analysis to meet our goals.

## Data Analytics Process:

1. Ask
2. Prepare
3. Process
4. Analyze
5. Share
6. Act

### Ask:

We decide which things and facilities we need in and around the house. Facilities like nearby hospitals, schools, vegetable markets, flooding, drainage system during rainy seasons, Size, Surrounding area, Interior of the house, and Parking Space.

### Prepare:

I am willing to spend 80 lakhs. I have 20 lakhs worth of fixed deposits in my bank and a bank balance of around 40 lakhs. And for the rest of the 20 lakhs, I am planning to take a loan from bank.

### Process:

I need to buy a three BHK house. Which has a garden and a big balcony.

### Analyse:

We are a family of four including Mom, Dad, Myself, and a younger brother so we need three separate bedrooms, one for mom & dad, one for myself, and one for my brother. Dad likes planting so having our personal garden is also necessary. We also have a car so a personal space for parking is also needed. I like to enjoy my evenings while looking at sunsets so a balcony is also needed. The interior should be clean and minimalistic which is in trend nowadays.

### Share:

After communicating the details to the housing agent that we need these specific things in a specific manner he showed us three houses of which the first one had very bright colors, the second one was a little out of our budget, but the third one fit all of our criteria and budget.

### Act:

Then we finally carried out the loan from the bank, made a deal, and purchased the house.

# Module 2: Instagram User Analytics

## Project Description:

The project is about the user analytics of Instagram. The project is to analyse the best possible outcomes to some known problems related to marketing and the performance of the Instagram. The data tables given for the project is users, photos, comments, tags, likes, follow, photo\_tags. MySQL is used to solve the problems.

## Approach:

My approach will be to first filter out the useful insights from the data on various parameters which will give an idea about how the marketing team can run their Ad campaign successfully and the investors can know how the app is performing. Firstly I observed all the given tables in the data and then wrote different queries to filter out and get the required data from all the tables.

## Tech-Stack Used:

To solve these problems, I used MySql because of its easy-to-understand and easy-to-use configuration

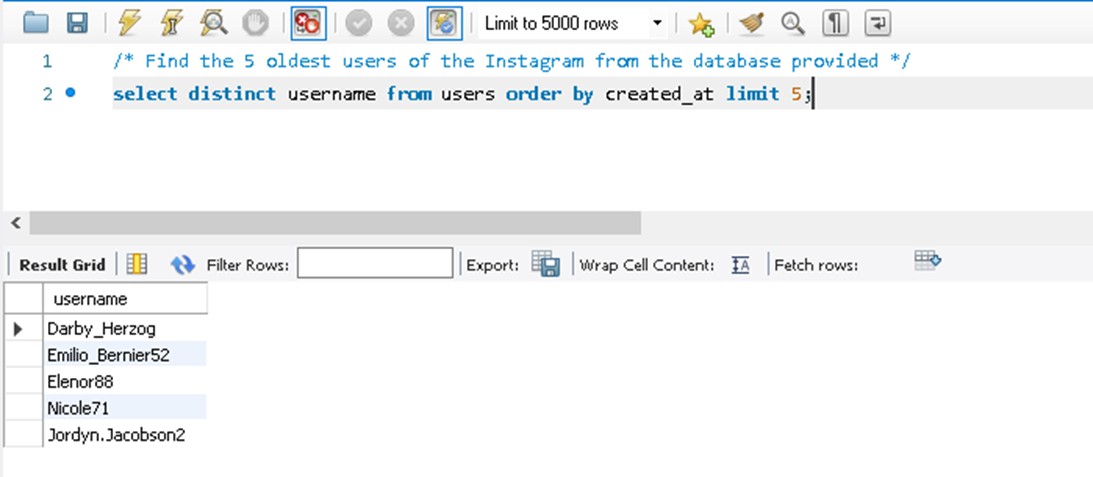
## Insights:

The project is extremely helpful to understand basics of SQL. It helped me to learn the structure. It also helped me to learn new keywords like dayofweek etc. I have also learned the concept of JOIN, HAVING, WHERE, IN, NOT IN, GROUP BY, ORDER BY, etc. This

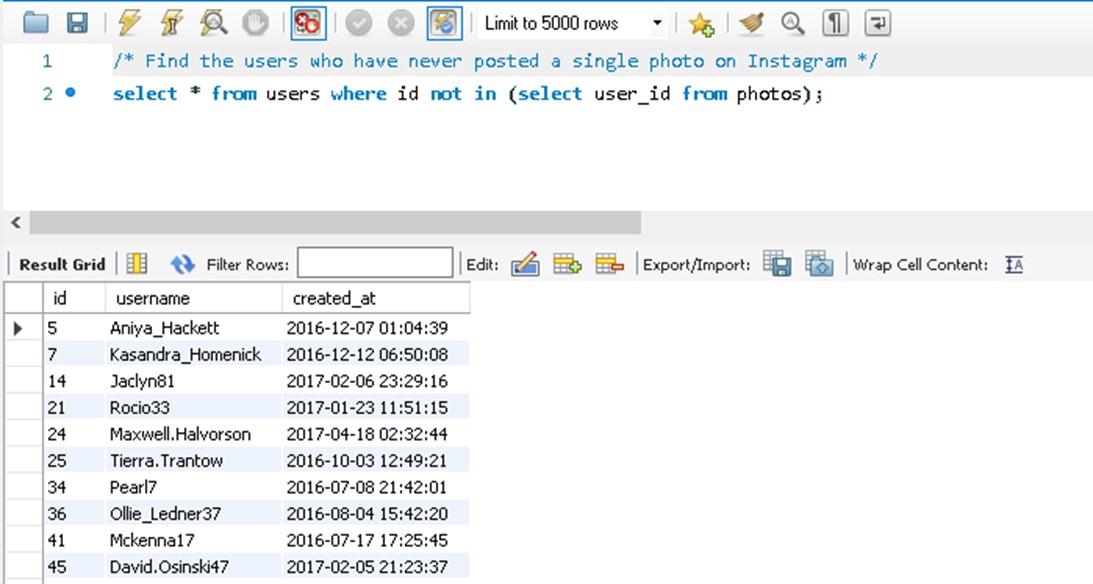
project gave me the confidence to work in SQL. I also learned how we can make use of past data to enhance the performance and efficiency of the current/future product of the company. By using data, we can run appropriate and successful marketing campaigns for the right set of customers.

## Result:

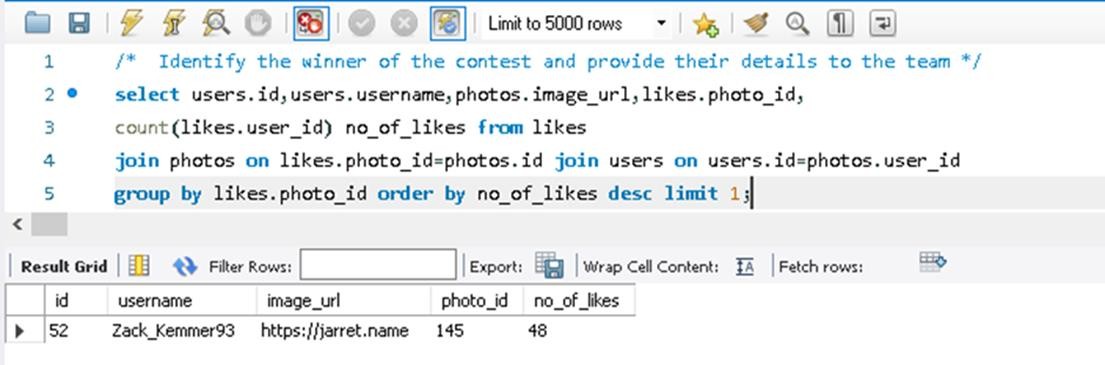
1. **Marketing:** The marketing team wants to launch some campaigns. Help them.
2. **Rewarding most loyal users:** People who have been using the platform for the longest time.



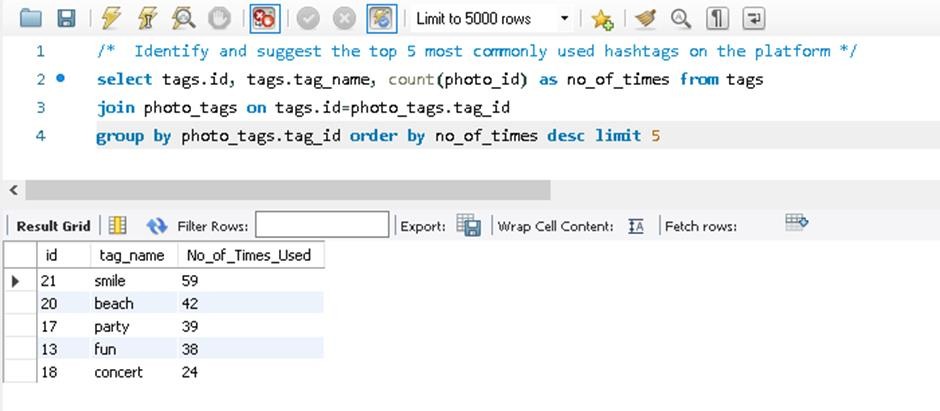
1. **Remind Inactive Users to Start Posting:** By sending them promotional emails to post their 1st photo.



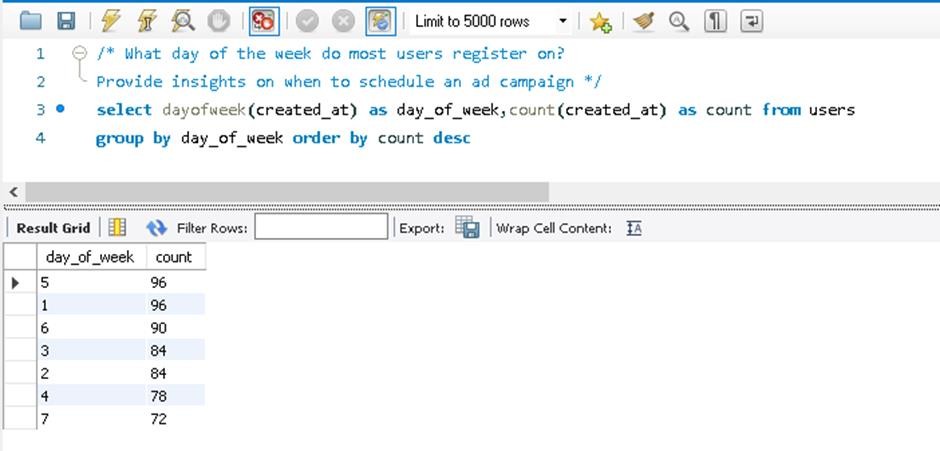
1. **Declaring Contest Winner:** The team started a contest and the user who gets the most likes on a single photo will win the contest now they wish to declare the winner.



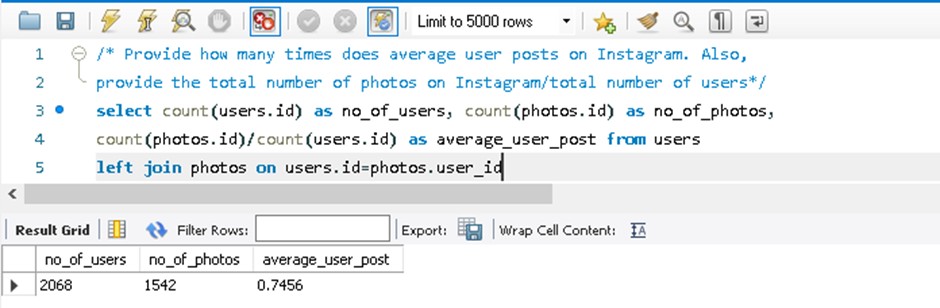
1. **Hashtag Researching:** A partner brand wants to know, which hashtags to use in the post to reach the most people on the platform.



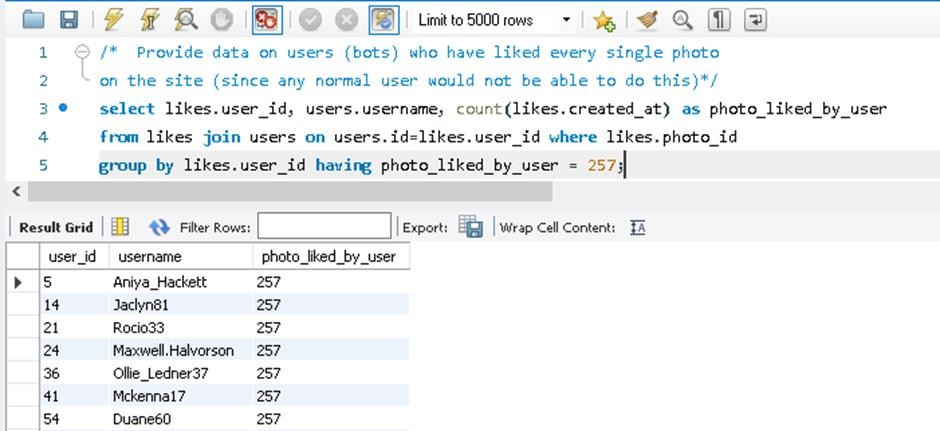
1. **Launch AD Campaign:** The team wants to know, which day would be the best day to launch ADs.



1. **Investor Metrics:**
2. **User Engagement:** Are users still as active and post on Instagram or they are making fewer posts.



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



# Module 3: Operation and Metric Analytics

## Description:

Operation analytics is the subset of data analytics that mainly focuses on the improving efficiency of the operations in the company. It shows how currently different operations are working and how those operations can be improved for better profits. In this project, I have answered such questions which are generally asked by the different departments in a company like the ops team, support team, marketing team, etc. which are required to increase efficiency and streamline.

Investigating metric spikes is an important part of operation analytics. The metric spike shows the anomaly in the trends. It gives the answers to questions like- why there is a dip in daily engagement. why have sales taken a dip? etc. These questions are to be answered daily or weekly basis and should be treated seriously. For this task, different data sets are given. The project is about using various sets of data to derive various operation analytics for the company so the company can find out the areas it needs to work on and improve to get better results.

## Approach:

The approach towards this project was simple and basic first study all the tables given in the dataset and then create a database and upload the data given into the database into the MySQL workbench then made necessary changes to the data like adding more rows where needed, then writing queries which will help filter out the data as per our need and get the needed insights to answer the questions of different teams

## Tech-Stack used:

To solve these problems, I have used MYSQL Workbench 8.0 CE. Which is an open software and can be downloaded from <https://www.mysql.com/>.

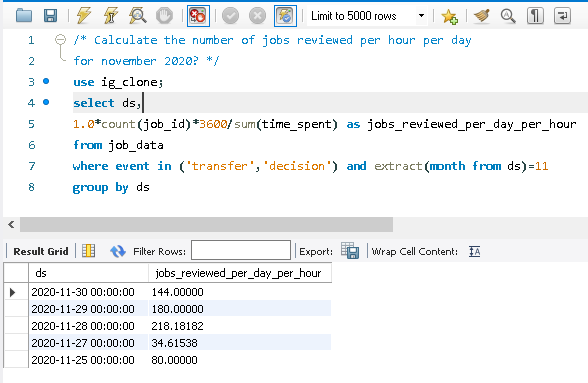
## Insights:

The project taught me that every company no matter how big or small has important questions about various departments of sales, marketing, etc. which are needed to be answered on a daily basis to track their progress. Every dip and rise in the company’s performance or of various teams in the company can be tracked and the reason behind it can be known through various data sets by manipulating the data and deriving useful insights from it.

## Results:

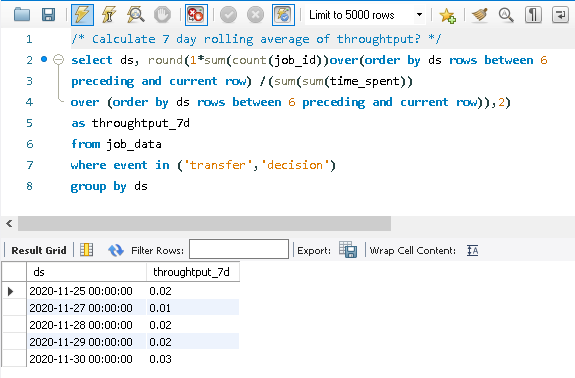
##### Case Study 1 (Job Data):

**A. Number of jobs reviewed:** Amount of jobs reviewed over time.

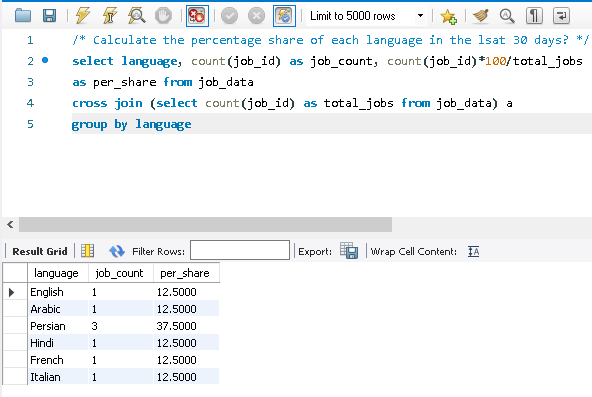


##### Throughput: It is the no. of events happening per second. Let’s say the above metric is called throughput. For throughput, do you prefer daily metric or 7-day rolling, and why?

##### If the density of the data is larger, we use the daily metric, and if the density is low the 7-day rolling works well. It also depends upon the anomaly in the data set. Because in 7 days metric the view is broader similar to the daily metric view becomes narrower.

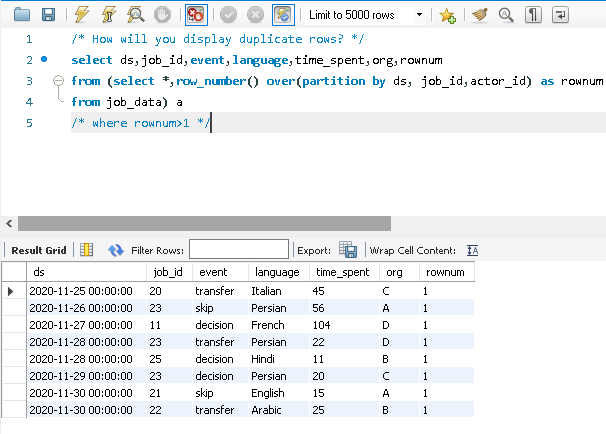


##### Percentage share of each language: Share of each language for different contents.



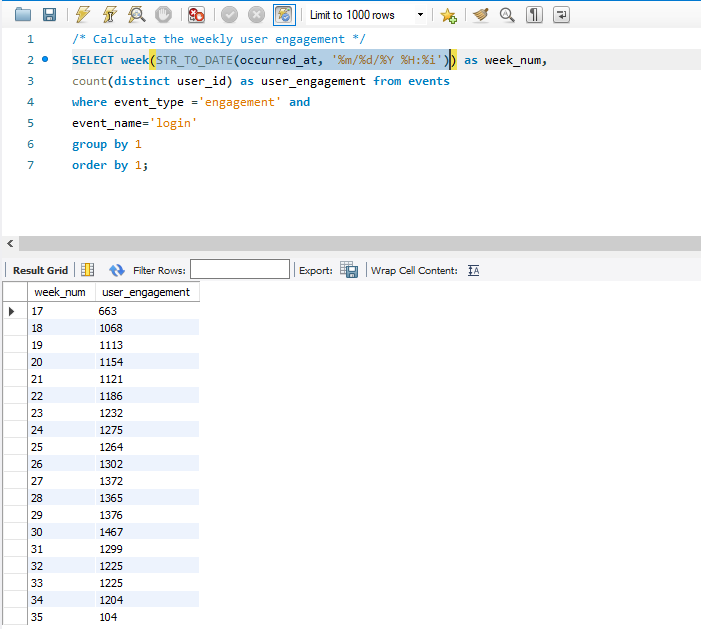
##### Duplicate rows: Rows that have the same value present in them.

##### When the row has been duplicated the value of the ‘rownum’ column in the output will be greater than 1. And the duplicate rows can be extracted by removing the comment in the where clause.

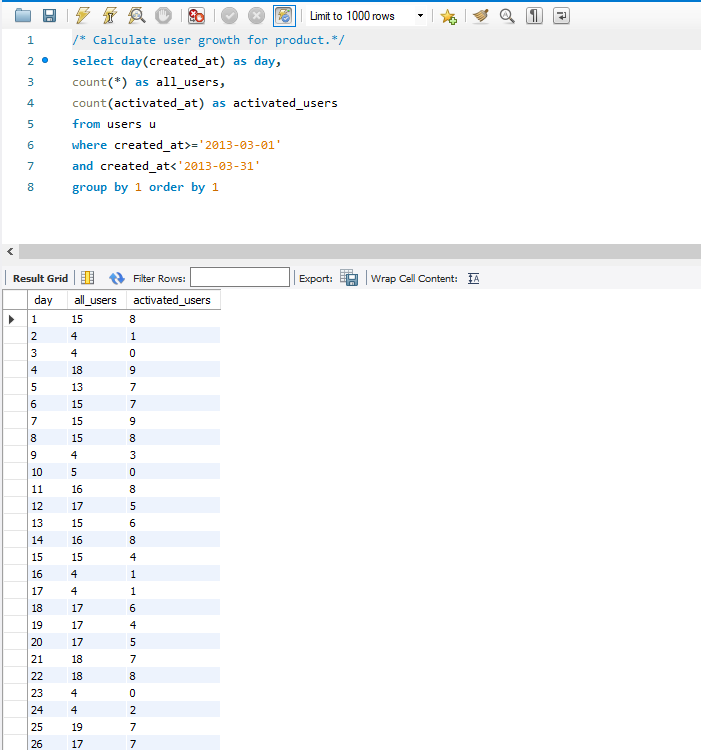


##### Case Study 2 (Investigating Metric Spike):

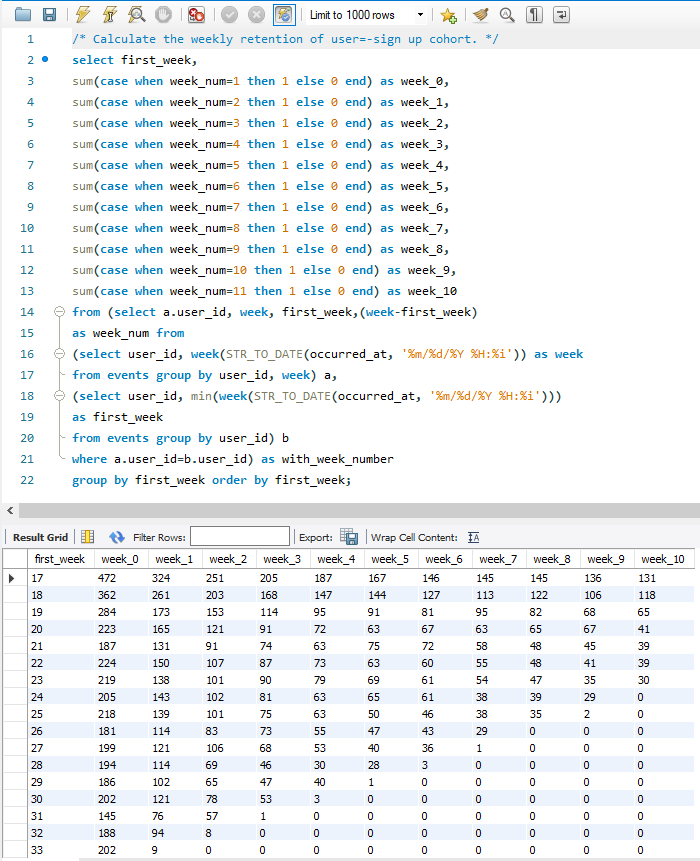
##### User Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service.



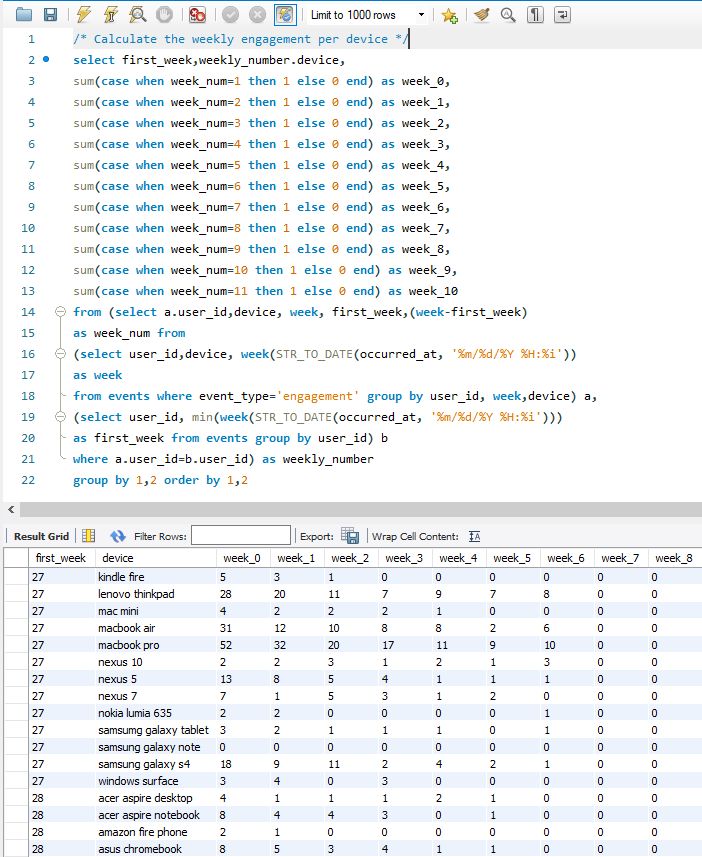
##### User Growth: Amount of users growing over time for a product.



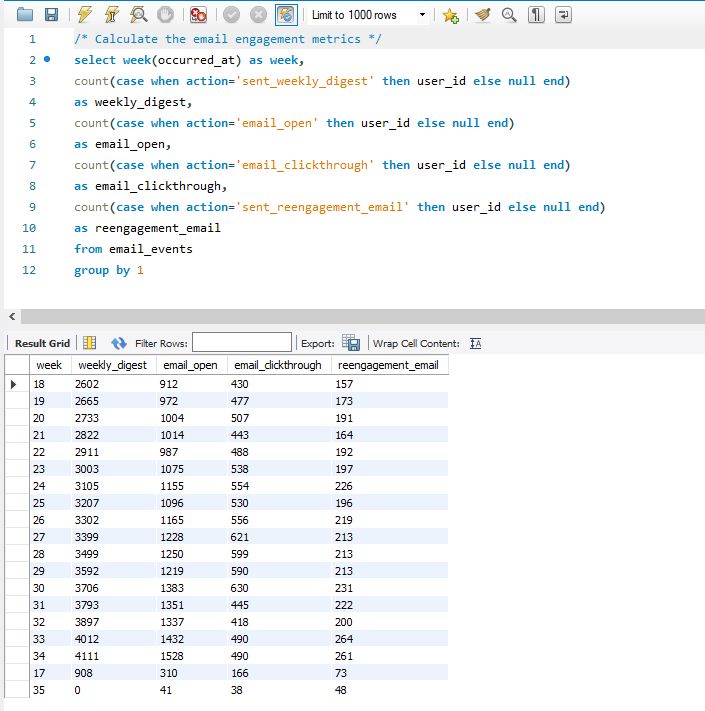
##### Weekly Retention: Users get retained weekly after signing up for a product.



##### Weekly Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.



##### Email Engagement: Users engaging with the email service.



# Module 4: Hiring Process Analytics

## Project Description:

Hiring process is the fundamental and the most important function of a company. Here, the MNCs get to know about the major underlying trends about the hiring process. Trends such as- number of rejections, number of interviews, types of jobs, vacancies etc. are important for a company to analyse before hiring freshers or any other individual.

The data set given is of a company where details about the people who registered for a particular post in a department of this company.

## Approach:

First, to understand the data set I have performed the EDA on the data set. Checked for the null values and the distribution. Now to answer each question I first understood all the questions. What should be outcome for each of the question and what columns to use. Now to get result I checked all the functions which will be required to perform the operations.

## Tech-Stack Used:

For this assignment I have used Microsoft Excel (2019).

## Insights:

This assignment helps me to understand how a company use hiring data to track the acceptance and rejection for a particular post. I have used pie chart, histogram, and bar plots. I have learned max, min, and average functions.

## Result:

* + 1. **Hiring:** Process of intaking of people into an organization for different kinds of positions.

**Your task:** How many males and females are Hired?

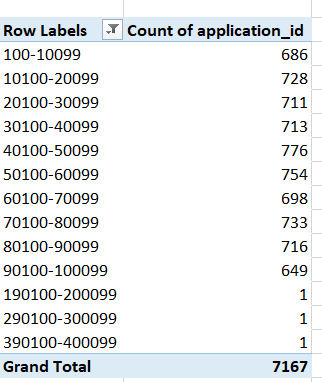
|  |  |
| --- | --- |
| Male | 4085 |
| Female | 2675 |

* + 1. **Average Salary:** Adding all the salaries for a select group of employees and then dividing the sum by the number of employees in the group. **Your task:** What is the average salary offered in this company?

|  |  |
| --- | --- |
| **Avg Salary** | **49983.02902** |

* + 1. **Class interval:** The class interval is the difference between the upper-class limit and the lower-class limit.

**Your task:** Draw the class intervals for salary in the company?



* + 1. **Charts and Plots:** This is one of the most important parts of analysis to visualize the data.

**Your task:** Draw Pie Chart / Bar Graph (or any other graph) to show proportion of

people working different department?

|  |  |
| --- | --- |
| **Row Labels** | **Count of**  **Department** |
| Finance Department | 288 |
| General Management | 172 |
| Human Resource Department | 97 |
| Marketing Department | 325 |
| Operations Department | 2771 |
| Production Department | 380 |
| Purchase Department | 333 |
| Sales Department | 747 |
| Service Department | 2055 |
| **Grand Total** | **7168** |

**2%**

**4%**

**Total**

**1%**

**5%**

**29%**

**10%**

**39%**

**5%**

**5%**

1141

710

535

405

400

288

232

251

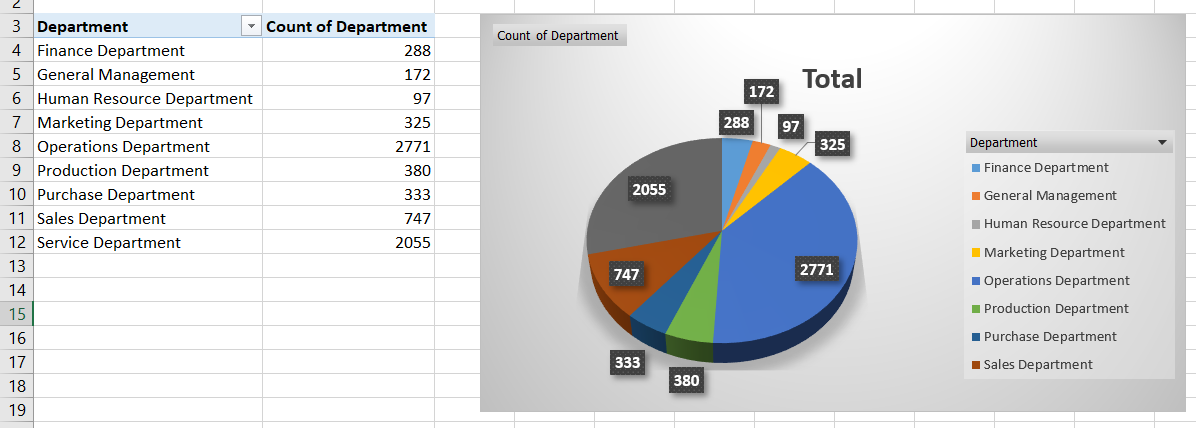
200 133 88

1 3 1 1 1 1

0

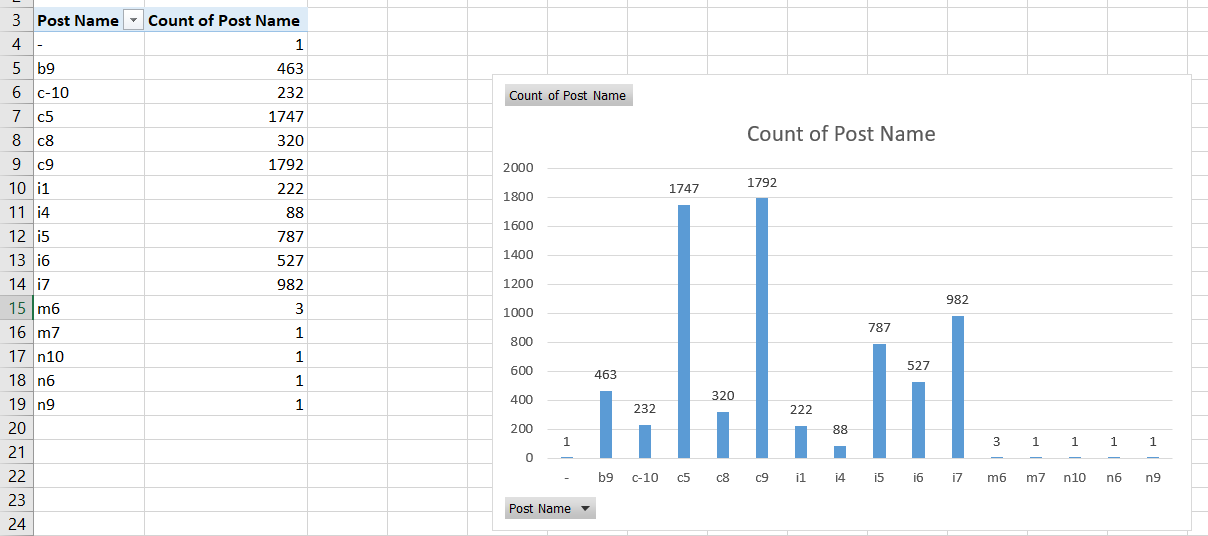
- b9 c-10 c5 c8 c9 i1 i4 i5 i6 i7 m6 m7 n10 n6 n9

POST NAME



* + 1. **Charts:** Use different charts and graphs to perform the task representing the data.

**Your task:** Represent different post tiers using chart/graph?



# Module 5: IMDB Movie Analysis

## Project Description:

The project is to analyse the movies trend, which movie is performing better, reason behind that, which movie is more popular, which actor is popular and its root cause. The data set given for this project consist of 28 columns and 5044 data points. Which contains actor names, IMDB rating, movie title, year, genres, director, reviews, language, budget, gross, etc. This also helps us to understand root cause analysis ‘Five Whys’ approach.

## Approach:

First download the data and analyse it by checking for the null values in each row as well as in each column, duplicate rows and in which column data can be interpolated. Identified for each of the question which approach to be best. Which column is most useful to answer the questions and the null values removed from it. For each of the result the chart is the best approach to present it. So, I created chart for each of the result.

## Tech-Stack Used:

To perform tasks for the project I have used Microsoft Excel (2019), Microsoft PowerPoint

(2019).

## Insights:

This assignment helps me to understand the functions in MS Excel and the pivot table. This gave me a complete idea of using the excel and the power of excel itself. Also, how a movie performance affect with the time, actor selection, director, reviews and IMDB rating.

## Results:

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

The total number of datapoint before cleaning was 5043.

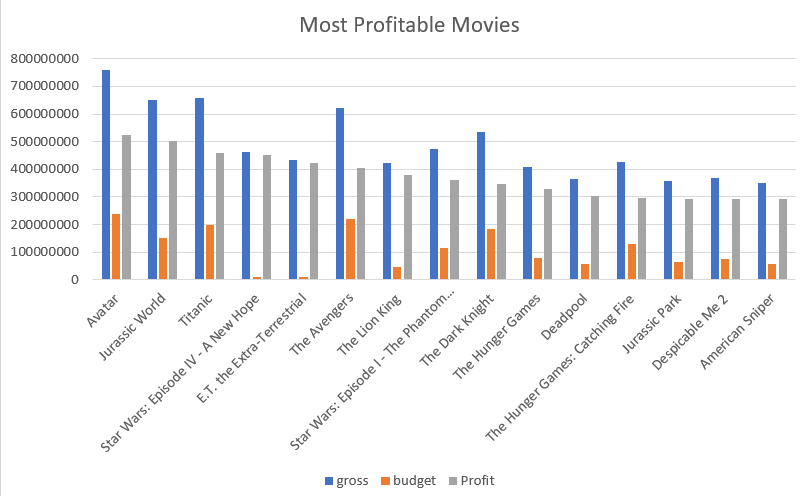
|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Step** | **Rows Removed** | **Rows Left** |
| 1 | Remove Duplicates | 45 | 4998 |
| 2 | Removes those rows which have more than 7 columns Null | 19 | 4979 |
| 3 | Remove rows where gross = Null | 859 | 4120 |
| 4 | Remove rows where budget = Null | 263 | 3857 |
| 5 | Replace ‘Null’ values in language column with ‘English’ | 3857 | 3857 |
| 6 | Removes those rows which have more than 3 columns Null | 6 | 3851 |

Rows left after the cleaning is 3851 which is 76.3% of the original data.

1. **Movies with highest profit:** Create a new column called profit which contains the difference of the two columns: gross and budget. Sort the column using the profit column as 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?

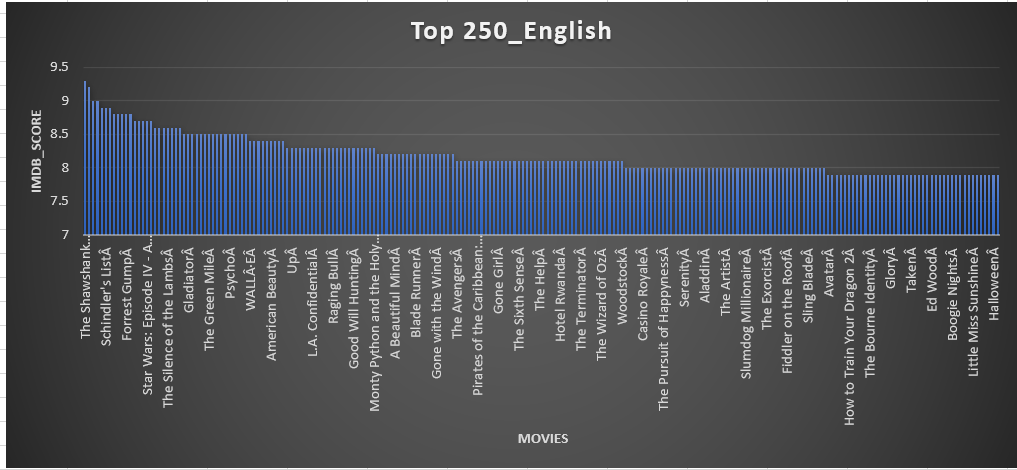
PROFIT

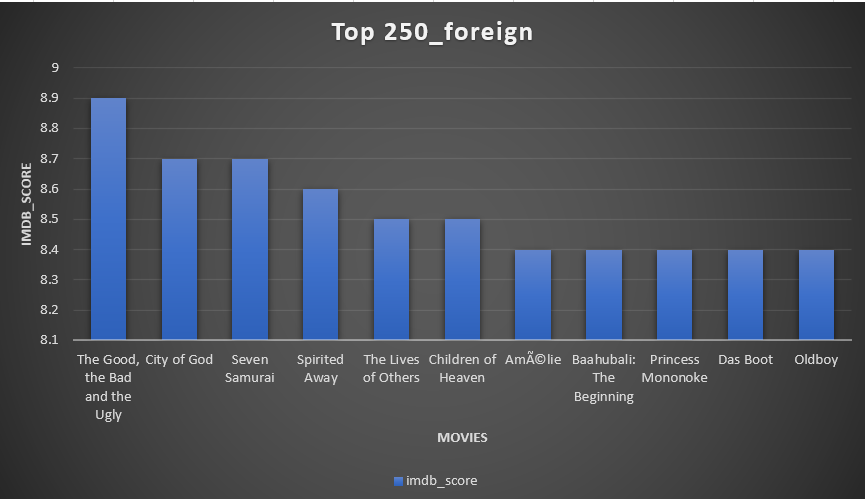


1. **Top 250:** Create a new column IMDb\_Top\_250 and store the top 250 movies with the highest IMDb Rating (corresponding to the column: imdb\_score). Also make sure that for all of these movies, the num\_voted\_users are greater than 25,000. Also add a Rank column containing the values 1 to 250 indicating 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 25

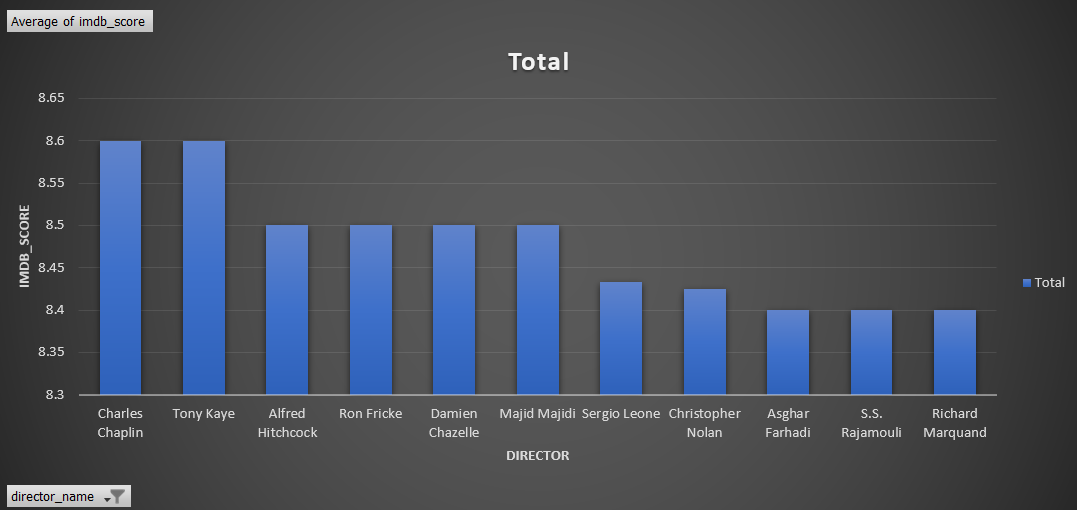




1. **Best Directors:** TGroup the column using the director\_name column.

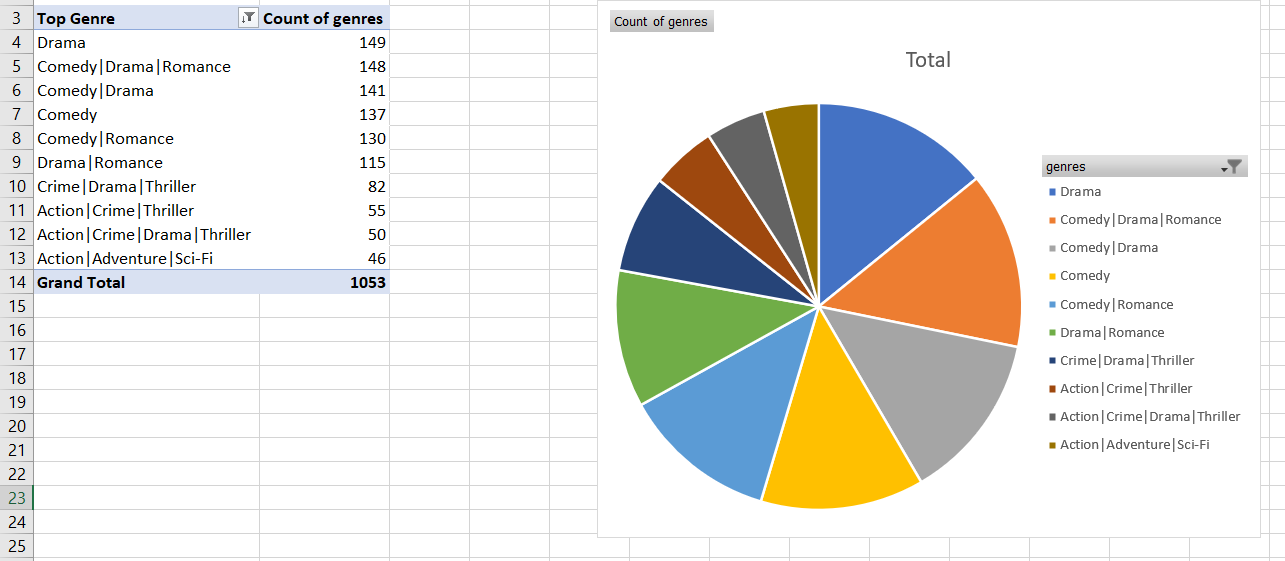
Find out the top 10 directors for whom the mean of imdb\_score is the highest and store them in a new column top10director. In case of a tie in IMDb score between two directors, sort them alphabetically

**Your task:** Find the best directors



**E. Popular Genres:** Perform this step using the knowledge gained while performing previous steps.

**Your task:** Find popular genres



* + 1. **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.

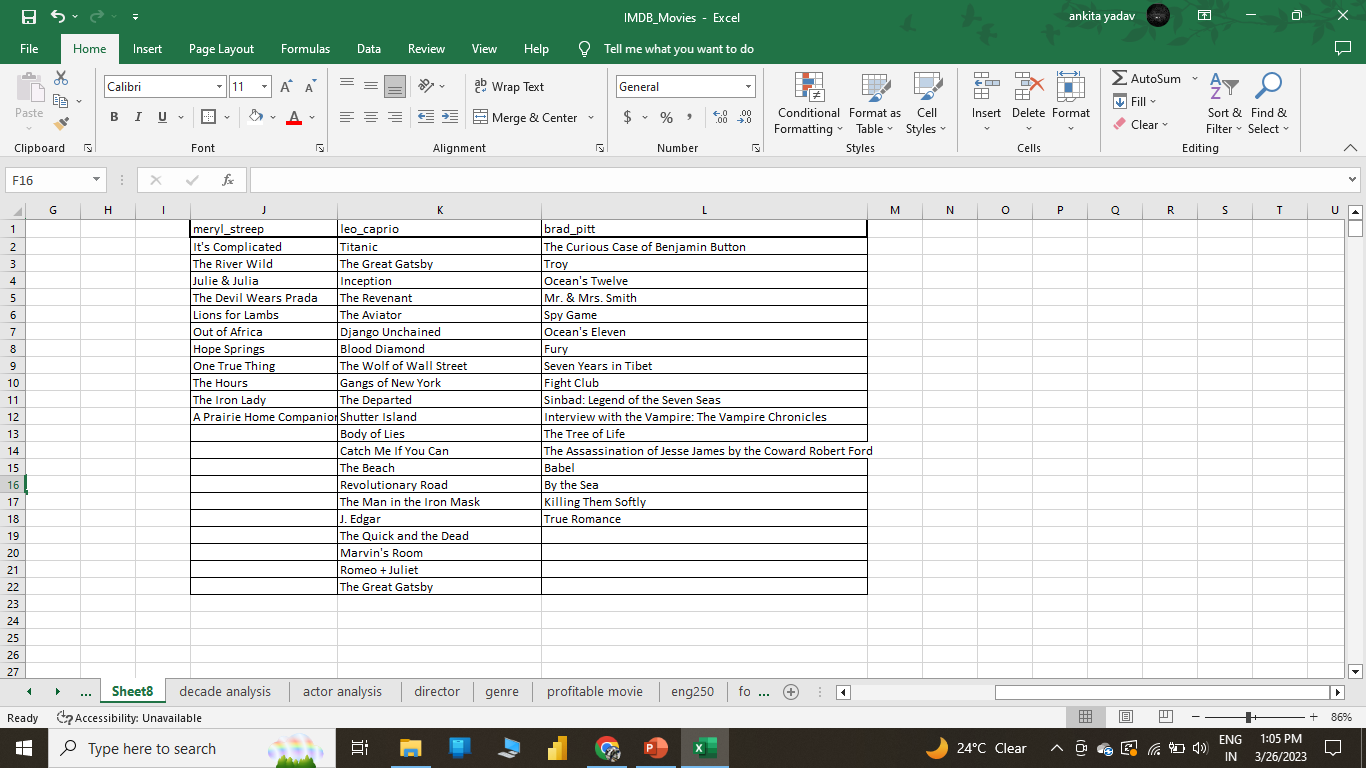
Append the rows of all these columns and store them in a new column named Combined.

Group the combined column using the actor\_1\_name column.

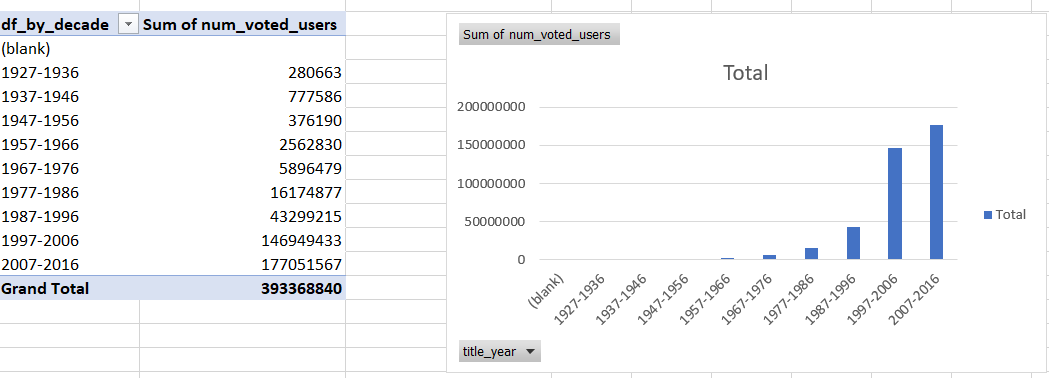
Find the mean of the num\_critic\_for\_reviews and num\_users\_for\_review and identify the actors which have the highest mean.

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.

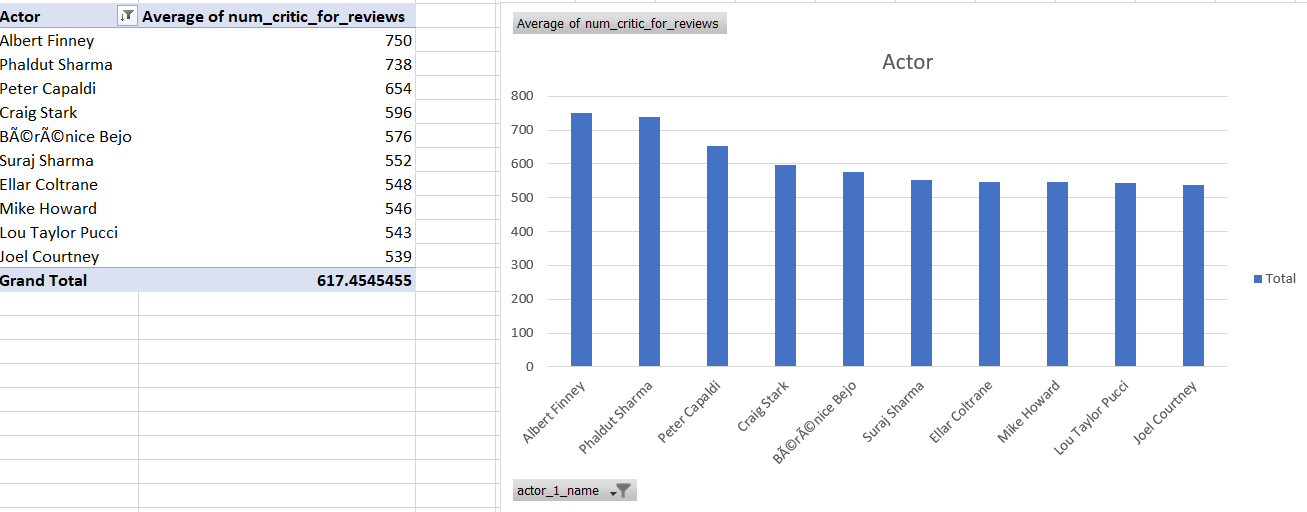
**Your task:** Find the critic-favorite and audience-favorite actors



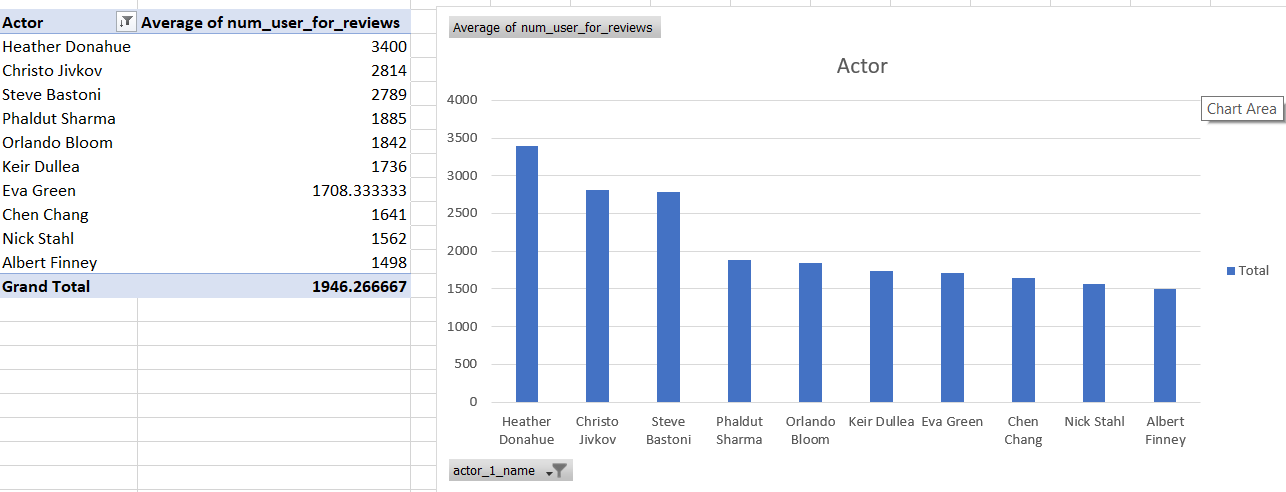
**Decade Analysis:**

****

**Critic Favorite Actor:**

****

**Audience Favorite Actor:**

****

## Conclusion:

Through this project I get to know how we can use different functions in different ways to manipulate the data as per our needs and derive various insights which show us important parameters. I have now gained good knowledge about various tools in functions in excel and the process of manipulating the data and using that to create various data visualizations to present the insights in a visually appealing manner.

**Description:**

# Module 6: Bank Loan Case Study

Bank earns their profits by providing loans to the customer. But whenever a person is not able to repay the loan bank suffer from the loss. Also, some of the people who be the defaulter apply for the loan again which will cost the bank. Analysing the defaulter in bank is a major work of risk analyst. In this case study I will try to find out the relation between the defaulter and the different features of the data set, so that bank can earn profit.

The loss to the bank can be in two ways:

* 1. If a person is not likely to repay the loan, then accept ion his loan request will lead to the loss of the bank.
  2. If the person is likely to repay the loan but now approving the loan will lead to the loss of the bank.

When a client applies for a loan, there are four types of decisions that could be taken by the client/company:

1. **Approved:** The company has approved loan application
2. **Cancelled:** The client cancelled the application sometime during approval. Either the client changed her/his mind about the loan or in some cases due to a higher risk of the client he received worse pricing which he did not want.
3. **Refused:** The company had rejected the loan (because the client does not meet their requirements etc.).
4. **Unused Offer:** Loan has been cancelled by the client but on different stages of the process.

## Approach:

First understanding both the dataset and what is the key connection between them. Check for the null values and plotting some scatter plots and bar plots to understanding the distribution. For each of the result the chart is the best approach to present it. So, I created chart for each of the result.

## Tech used:

For the project I have used Microsoft Excel and Jupyter Notebook.

## Insights:

This project helps me to understand the power of different platforms. I understood that for the large data set where the total number of data point higher than 10 Lac it is better to use Python. Also, with this project I learned how to deal with a lot of features in a data set.

I have also find worked out on the following problems:

* Present the overall approach of the analysis. Mention the problem statement and the analysis approach briefly
* Identify the missing data and use appropriate method to deal with it. (Remove columns/or replace it with an appropriate value) Hint: Note that in EDA, since it is not necessary to replace the missing value, but if you have to replace the missing value, what should be the approach. Clearly mention the approach.
* Identify if there are outliers in the dataset. Also, mention why do you think it is an outlier. Again, remember that for this exercise, it is not necessary to remove any data points.
* Identify if there is data imbalance in the data. Find the ratio of data imbalance. Hint: Since there are a lot of columns, you can run your analysis in loops for the appropriate columns and find the insights.
* Explain the results of univariate, segmented univariate, bivariate analysis, etc. in business terms.
* Find the top 10 correlation for the Client with payment difficulties and all other cases (Target variable). Note that you have to find the top correlation by segmenting the data frame w.r.t to the target variable and then find the top correlation for each of the segmented data and find if any insight is there. Say, there are 5+1(target) variables in a dataset: Var1, Var2, Var3, Var4, Var5, Target. And if you have to find the top 3 correlations, it can be: Var1 & Var2, Var2 & Var3, Var1 & Var3. Target variable will not feature in this correlation as it is a categorical variable and not a continuous variable that is increasing or decreasing.
* Include visualizations and summarize the most important results in the presentation. You are free to choose the graphs which explain the numerical/categorical variables. Insights

should explain why the variable is important for differentiating the clients with payment difficulties with all other cases.

## Results:

1. **Importing libraries:**

import pandas as pd

import matplotlib.pyplot as plt import seaborn as sns sns.set\_theme(style="whitegrid")

import numpy as np

1. **Creating function to find the null values in ascending order**

def count\_null\_values(df,per=0): length = len(df)

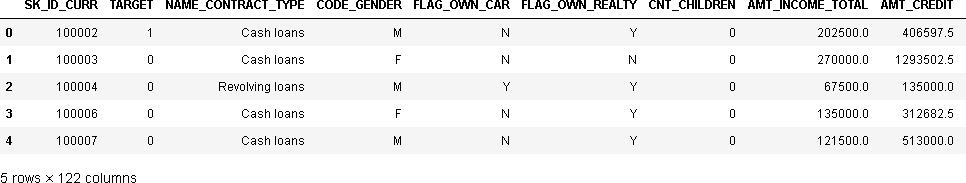
df1 = df.isnull().sum()\*100/length df2 = df1[df1>=per]

return df2.sort\_values(ascending=False)

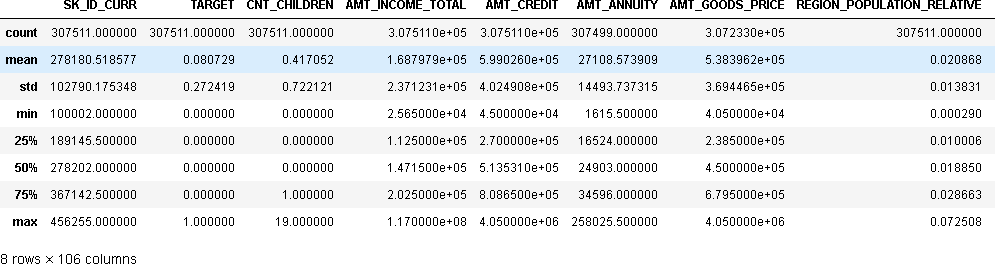
1. **Reading the First dataset (application.csv):**

*# application\_data.csv*

df1 = pd.read\_csv('/content/drive/MyDrive/Trainity/Loan Case Study/application\_data.csv') df1.head()



1. **Find the description of the dataset:**

 df1.describe()

1. **Dealing with the null values:**

# Check for the null values greater than 50% in each column in df1

null50 = count\_null\_values(df1,50)

null50 ['OWN\_CAR\_AGE',

'EXT\_SOURCE\_1', 'APARTMENTS\_AVG', 'BASEMENTAREA\_AVG', 'YEARS\_BUILD\_AVG', 'COMMONAREA\_AVG', 'ELEVATORS\_AVG', 'ENTRANCES\_AVG', 'FLOORSMIN\_AVG', 'LANDAREA\_AVG', 'LIVINGAPARTMENTS\_AVG', 'LIVINGAREA\_AVG', 'NONLIVINGAPARTMENTS\_AVG', 'NONLIVINGAREA\_AVG', 'APARTMENTS\_MODE', 'BASEMENTAREA\_MODE', 'YEARS\_BUILD\_MODE', 'COMMONAREA\_MODE', 'ELEVATORS\_MODE', 'ENTRANCES\_MODE', 'FLOORSMIN\_MODE', 'LANDAREA\_MODE', 'LIVINGAPARTMENTS\_MODE', 'LIVINGAREA\_MODE', 'NONLIVINGAPARTMENTS\_MODE', 'NONLIVINGAREA\_MODE', 'APARTMENTS\_MEDI', 'BASEMENTAREA\_MEDI', 'YEARS\_BUILD\_MEDI', 'COMMONAREA\_MEDI', 'ELEVATORS\_MEDI', 'ENTRANCES\_MEDI', 'FLOORSMIN\_MEDI', 'LANDAREA\_MEDI', 'LIVINGAPARTMENTS\_MEDI', 'LIVINGAREA\_MEDI', 'NONLIVINGAPARTMENTS\_MEDI', 'NONLIVINGAREA\_MEDI', 'FONDKAPREMONT\_MODE', 'HOUSETYPE\_MODE', 'WALLSMATERIAL\_MODE']

**Inference:**

The null columns contain the values most related to the apartment or house, and one column contain data for car age, and one normalized score for external data source. So that can be removed because they cannot be useful for the target.

##### # drop values with more than 50% null values

df1.drop(null50.index,axis=1,inplace=True) df1.shape

(307511, 81)

##### # Check for the null values greater than 15% in each column in df1

print('Percentage (%) null values in each column') null15 = count\_null\_values(df1,15)

Percentage (%) null values in each column FLOORSMAX\_AVG 49.760822

FLOORSMAX\_MODE 49.760822

FLOORSMAX\_MEDI 49.760822

YEARS\_BEGINEXPLUATATION\_AVG 48.781019

YEARS\_BEGINEXPLUATATION\_MODE 48.781019

YEARS\_BEGINEXPLUATATION\_MEDI 48.781019

|  |  |  |
| --- | --- | --- |
| TOTALAREA\_MODE | | 48.268517 |
| EMERGENCYSTATE\_MODE | | 47.398304 |
| OCCUPATION\_TYPE | | 31.345545 |
| EXT\_SOURCE\_3 | 19.825307 | |
| DAYS\_EMPLOYED  dtype: float64 | 18.007161 | |

**Inference:**

In this the except occupation\_type and ext\_souorce\_3 looks familiar to target null values and those are related to the building. So that can be removed.

##### # Remove those columns and drop remaining columns

columns\_with\_null = dict(null15)

del columns\_with\_null['EXT\_SOURCE\_3']

del columns\_with\_null['OCCUPATION\_TYPE'] df1.shape

(307511, 72)

1. **Feature selection and remove unnecessary columns**

##### # Treating the columns with the null values

null\_count = count\_null\_values(df1) null\_count

OCCUPATION\_TYPE 31.345545

EXT\_SOURCE\_3 19.825307

AMT\_REQ\_CREDIT\_BUREAU\_DAY 13.501631

AMT\_REQ\_CREDIT\_BUREAU\_HOUR 13.501631

AMT\_REQ\_CREDIT\_BUREAU\_YEAR 13.501631

...

REG\_REGION\_NOT\_WORK\_REGION 0.000000

LIVE\_REGION\_NOT\_WORK\_REGION 0.000000

REG\_CITY\_NOT\_LIVE\_CITY 0.000000

TARGET 0.000000

REG\_CITY\_NOT\_WORK\_CITY 0.000000

Length: 72, dtype: float64

##### # check correlation for the columns EXT\_SOURCE\_3 and EXT\_SOURCE\_2 to the targ et column

sns.heatmap(df1[['EXT\_SOURCE\_3','EXT\_SOURCE\_2','TARGET']].corr(),annot=True)



**Inference:**

As it is clearly seen that the both columns do not show a good correlation with the Target. So, it can be removed.

df1.drop(['EXT\_SOURCE\_3','EXT\_SOURCE\_2'],axis=1,inplace=True) df1.shape

(307511, 70)

##### # Now there are some columns named *flag* which contain some true false values

flags = []

for i in df1.columns: if 'FLAG' in i: flags.append(i)

flags

['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']

##### # Plot each flag column vs target column # Plot will be of 28 plots with hue = Y/N

plt.figure(figsize=(30,40))

for i in range(len(flag\_df.columns)-1):

plt.subplot(7,4,i+1) sns.countplot(x=flag\_df[flag\_df.columns[i]],hue=flag\_df[flag\_df.columns[-1]])



**Inference:**

All the columns which have sufficient values in both 0 and 1 will be kept, except removed columns to kept FLAG\_OWN\_CAR, FLAG\_OWN\_REALTY, FLAG\_PHONE, FLAG\_WORK\_PHONE.

##### # Drop unnecessary flag columns:

remove\_list = ['FLAG\_OWN\_REALTY','FLAG\_OWN\_CAR','FLAG\_PHONE','FLAG\_WORK\_PHONE'

]

flags = list(set(flags) - set(remove\_list)) # finally remove those columns df1.drop(flags,axis=1,inplace=True)

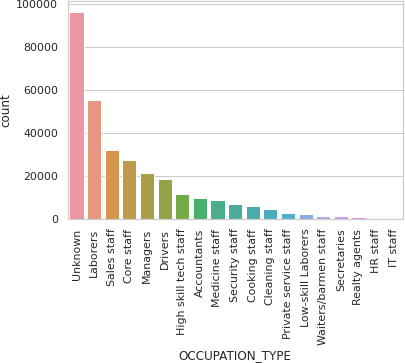
# Dataset after removing unnecessary flag values

df1.shape

(307511, 46)

1. **Filling null values**

##### # Fill the null values in OCCUPATION\_TYPE with the 'Unknown'



df1['OCCUPATION\_TYPE'].fillna('Unknown',inplace=True)

sns.countplot(x =df1['OCCUPATION\_TYPE'],order = df1['OCCUPATION\_TYPE'].value\_counts().index

)

plt.xticks(rotation=90)

##### # Filling values of the column name starts with amt\_req

amt\_req = []

for i in df1.columns: if "AMT\_REQ" in i:

amt\_req.append(i) amt\_req

['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']

**Inference:**

The columns represent the Number of enquiries to Credit Bureau about the client before different time periods. these values should be integer.

# Mean

df1[amt\_req].mean() AMT\_REQ\_CREDIT\_BUREAU\_HOUR 0.006402

AMT\_REQ\_CREDIT\_BUREAU\_DAY 0.007000

AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0.034362

AMT\_REQ\_CREDIT\_BUREAU\_MON 0.267395

AMT\_REQ\_CREDIT\_BUREAU\_QRT 0.265474

AMT\_REQ\_CREDIT\_BUREAU\_YEAR 1.899974

dtype: float64

# Mode

df1[amt\_req].mode()



# Median

df1[amt\_req].median() AMT\_REQ\_CREDIT\_BUREAU\_HOUR 0.0

AMT\_REQ\_CREDIT\_BUREAU\_DAY 0.0

AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0.0

AMT\_REQ\_CREDIT\_BUREAU\_MON 0.0

AMT\_REQ\_CREDIT\_BUREAU\_QRT 0.0

AMT\_REQ\_CREDIT\_BUREAU\_YEAR 1.0

dtype: float64

##### # Filling the null values by Median

Inference:

For mean the values are real numbers so it cannot bs used, most values in all the columns are 0 but for the column AMT\_REQ\_CREDIT\_BUREAU\_YEAR occurrence of all the values is quite comparable so mode will not a good approach for all the columns, but for median it is

giving feasible values. So, I will replace all the values by median.

try1 = df1[amt\_req].fillna(df1[amt\_req].median())

**# For the column NAME\_TYPE\_SUITE I will replace the null values by Unknown**

# replace null by Unknown df1['NAME\_TYPE\_SUITE'].fillna('Unknown',inplace=True)

# Fill the null values for the column social circle

social\_circle = []

for i in df1.columns:

if 'SOCIAL\_CIRCLE' in i:

social\_circle.append(i) social\_circle

['OBS\_30\_CNT\_SOCIAL\_CIRCLE', 'DEF\_30\_CNT\_SOCIAL\_CIRCLE',

'OBS\_60\_CNT\_SOCIAL\_CIRCLE', 'DEF\_60\_CNT\_SOCIAL\_CIRCLE']

# Mean

df1[social\_circle].mean() OBS\_30\_CNT\_SOCIAL\_CIRCLE 1.422245

DEF\_30\_CNT\_SOCIAL\_CIRCLE 0.143421

OBS\_60\_CNT\_SOCIAL\_CIRCLE 1.405292

DEF\_60\_CNT\_SOCIAL\_CIRCLE 0.100049

dtype: float64

# Mode

df1[social\_circle].mode()



df1[social\_circle].median() OBS\_30\_CNT\_SOCIAL\_CIRCLE 0.0

DEF\_30\_CNT\_SOCIAL\_CIRCLE 0.0

OBS\_60\_CNT\_SOCIAL\_CIRCLE 0.0

DEF\_60\_CNT\_SOCIAL\_CIRCLE 0.0

dtype: float64

**Inference:**

This column represents how many observations of client's social surroundings day past due, which is integer. So, I will replace it by median.

# Filling values by median

df1[social\_circle] = df1[social\_circle].fillna(df1[social\_circle].median())

**# Fill the null values for AMT\_GOODS\_PRICE with the mean value**

df1['AMT\_GOODS\_PRICE'].fillna(df1['AMT\_GOODS\_PRICE'].mean(),inplace=True)

##### # Fill the null values for AMT\_ANNUITY with the mean value

df1['AMT\_ANNUITY'].fillna(df1['AMT\_ANNUITY'].mean(),inplace=True)

**# Filling CNT\_FAM\_MEMBERS with the median**

df1['CNT\_FAM\_MEMBERS'].fillna(df1['CNT\_FAM\_MEMBERS'].median(),inplace=True)

**# Filling CNT\_FAM\_MEMBERS with the median**

df1['DAYS\_LAST\_PHONE\_CHANGE'].fillna(df1['DAYS\_LAST\_PHONE\_CHANGE'].median(),inplace

=True)

**# find the columns with days**

days = []

for i in df1.columns: if 'DAYS' in i: days.append(i)

days

['DAYS\_BIRTH', 'DAYS\_REGISTRATION', 'DAYS\_ID\_PUBLISH',

'DAYS\_LAST\_PHONE\_CHANGE']

##### #There are 5 columns with the days change it in years

# change days to years df1[days] = abs(df1[days])/365 # Rename the columns

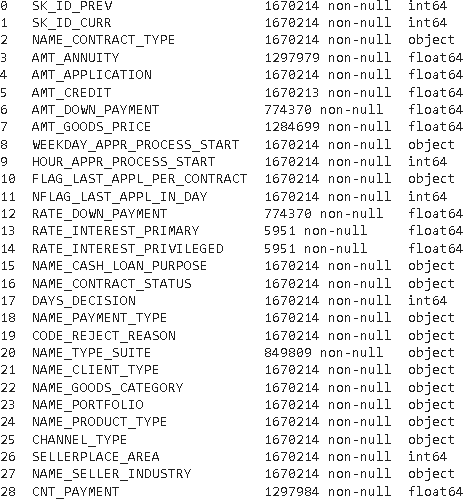
df1.rename(columns = {'DAYS\_BIRTH':'AGE','DAYS\_EMPLOYED':'YEARS\_EMPLOYED','DAYS\_R EGISTRATION':'YEARS\_REGISTRATION','DAYS\_ID\_PUBLISH':'YEARS\_ID\_PUBLISH','DAYS\_L

AST\_PHONE\_CHANGE':'YEARS\_LAST\_PHONE\_CHANGE'},inplace=True)

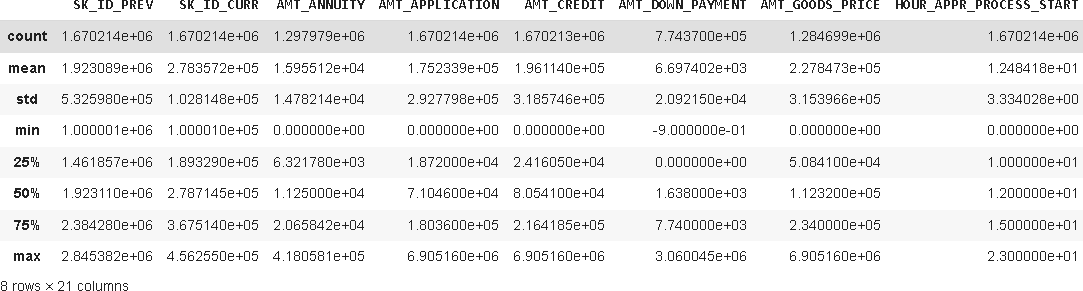
1. **Reading the dataset previous\_applications.csv**

df = pd.read\_csv('/content/drive/MyDrive/Trainity/Loan Case Study/previous\_application.csv')

df.info()



# Description of the data set

 df.describe()

1. **Dealing with null values**

# Count null values

count\_null\_values(df) RATE\_INTEREST\_PRIVILEGED 99.643698

RATE\_INTEREST\_PRIMARY 99.643698

AMT\_DOWN\_PAYMENT 53.636480

RATE\_DOWN\_PAYMENT 53.636480

NAME\_TYPE\_SUITE 49.119754

NFLAG\_INSURED\_ON\_APPROVAL 40.298129

DAYS\_TERMINATION 40.298129

DAYS\_LAST\_DUE 40.298129

DAYS\_LAST\_DUE\_1ST\_VERSION 40.298129

DAYS\_FIRST\_DUE 40.298129

DAYS\_FIRST\_DRAWING 40.298129

AMT\_GOODS\_PRICE 23.081773

AMT\_ANNUITY 22.286665

CNT\_PAYMENT 22.286366

PRODUCT\_COMBINATION 0.020716

AMT\_CREDIT 0.000060

NAME\_YIELD\_GROUP 0.000000

NAME\_PORTFOLIO 0.000000

NAME\_SELLER\_INDUSTRY 0.000000

SELLERPLACE\_AREA 0.000000

CHANNEL\_TYPE 0.000000

NAME\_PRODUCT\_TYPE 0.000000

SK\_ID\_PREV 0.000000

NAME\_GOODS\_CATEGORY 0.000000

NAME\_CLIENT\_TYPE 0.000000

CODE\_REJECT\_REASON 0.000000

SK\_ID\_CURR 0.000000

DAYS\_DECISION 0.000000

NAME\_CONTRACT\_STATUS 0.000000

NAME\_CASH\_LOAN\_PURPOSE 0.000000

NFLAG\_LAST\_APPL\_IN\_DAY 0.000000

FLAG\_LAST\_APPL\_PER\_CONTRACT 0.000000

HOUR\_APPR\_PROCESS\_START 0.000000

WEEKDAY\_APPR\_PROCESS\_START 0.000000

AMT\_APPLICATION 0.000000

NAME\_CONTRACT\_TYPE 0.000000

NAME\_PAYMENT\_TYPE 0.000000

dtype: float64

# Shape of the data set

df.shape (1670214, 37)

# Find the columns with more than 40% of null values

# find columns with more than 40% of null values #print('Percentage (%) null values in each column') null40 = count\_null\_values(df,40)

# this columns will be removed because of greater than 50% of null values

null40

RATE\_INTEREST\_PRIMARY 99.643698

RATE\_INTEREST\_PRIVILEGED 99.643698

AMT\_DOWN\_PAYMENT 53.636480

RATE\_DOWN\_PAYMENT 53.636480

NAME\_TYPE\_SUITE 49.119754

DAYS\_FIRST\_DRAWING 40.298129

DAYS\_FIRST\_DUE 40.298129

DAYS\_LAST\_DUE\_1ST\_VERSION 40.298129

DAYS\_LAST\_DUE 40.298129

DAYS\_TERMINATION 40.298129

NFLAG\_INSURED\_ON\_APPROVAL 40.298129

dtype: float64

# Dop all the values with more than 40% of null values

df.drop(null40,axis=1,inplace=True) df.shape

(1670214, 22)

# Remove the columns which are unnecessary

df.drop(['WEEKDAY\_APPR\_PROCESS\_START','HOUR\_APPR\_PROCESS\_START','FLAG\_LAST\_A

PPL\_PER\_CONTRACT','NFLAG\_LAST\_APPL\_IN\_DAY'],axis=1,inplace=True) df.shape

(1670214, 22)

# Now find the columns with missing values more than 15%

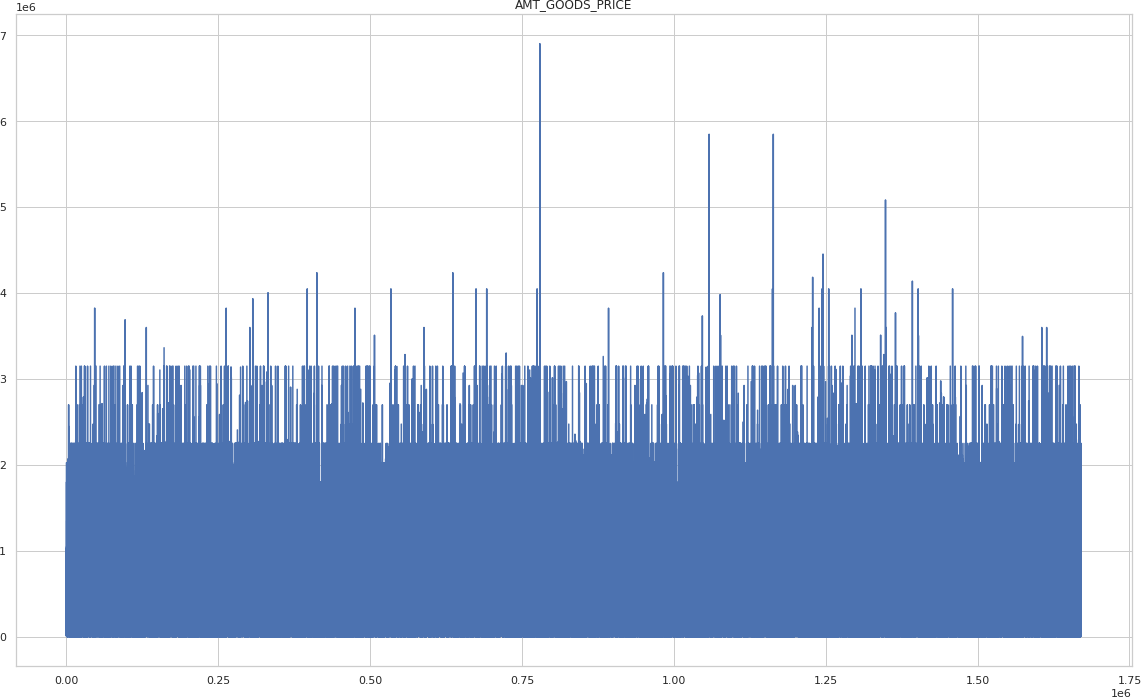
null15 = count\_null\_values(df,15)

# 10 columns with null values greater than 15% null15

['AMT\_ANNUITY', 'AMT\_GOODS\_PRICE', 'CNT\_PAYMENT']

# Plot goods price to check distributions

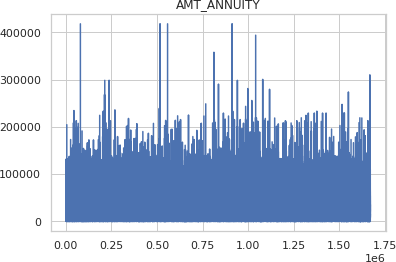
plt.figure(figsize=(20,12)) plt.plot(df['AMT\_GOODS\_PRICE'])



# Replace null in AMT\_GOODS\_PRICE by its mean as mean shows good replacement by the plot df['AMT\_GOODS\_PRICE'].fillna(df['AMT\_GOODS\_PRICE'].mean(),inplace=True)

# Plot amt annuity

plt.plot(df['AMT\_ANNUITY']) plt.title('AMT\_ANNUITY')

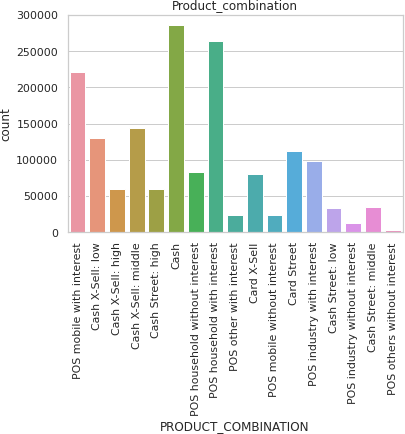


# replace null values in ANT\_ANNUITY by the mean

df['AMT\_ANNUITY'].fillna(df['AMT\_ANNUITY'].mean(),inplace=True)

# replace null values in CNT\_PAYMENT by its median df['CNT\_PAYMENT'].fillna(df['CNT\_PAYMENT'].median(),inplace=True)

# Plot product combination



sns.countplot(df['PRODUCT\_COMBINATION']) plt.xticks(rotation=90) plt.title(‘Product\_combination’)

# replace null with the Unknown in PRODUCT\_COMBINATION df['PRODUCT\_COMBINATION'].fillna('Unknown',inplace=True)

# Replace null in AMT\_CREDIT by its mean

df['AMT\_CREDIT'].fillna(df['AMT\_CREDIT'].mean(),inplace=True) # convert days to years for DAYS\_DECISION column

df['DAYS\_DECISION'] = abs(df['DAYS\_DECISION'])/365

1. **Check Outliers**
2. *application\_data.csv*

# Find outliers

out\_check = 'CNT\_CHILDREN,AMT\_INCOME\_TOTAL,AMT\_CREDIT,AMT\_ANNUITY,AMT\_GOO

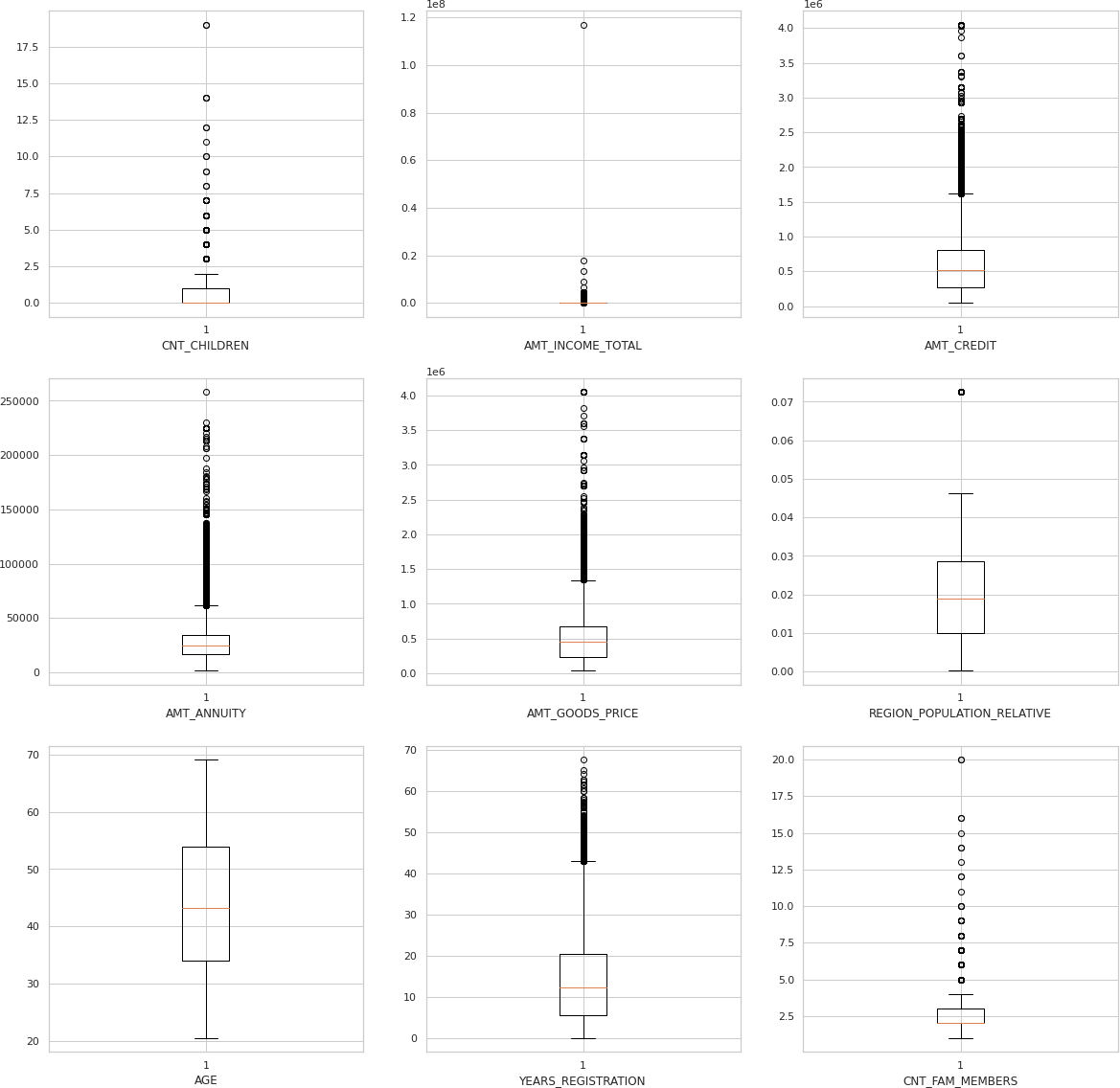
DS\_PRICE,REGION\_POPULATION\_RELATIVE,AGE,YEARS\_REGISTRATION,CNT\_FAM\_MEMB

ERS'.split(',') plt.figure(figsize=(20,20))

#df1.boxplot(column = out\_check, grid=False, rot=90, fontsize=15) for i in range(len(out\_check)):

plt.subplot(3,3,i+1) plt.boxplot(df1[out\_check[i]])

plt.xlabel(out\_check[i])



**Inference:**

There are outliers in each of the column except AGE. In the REGION\_POPULATION\_RELATIVE there are few outliers.

1. *Previous\_application.csv*

##### # Plot for outlier’s check

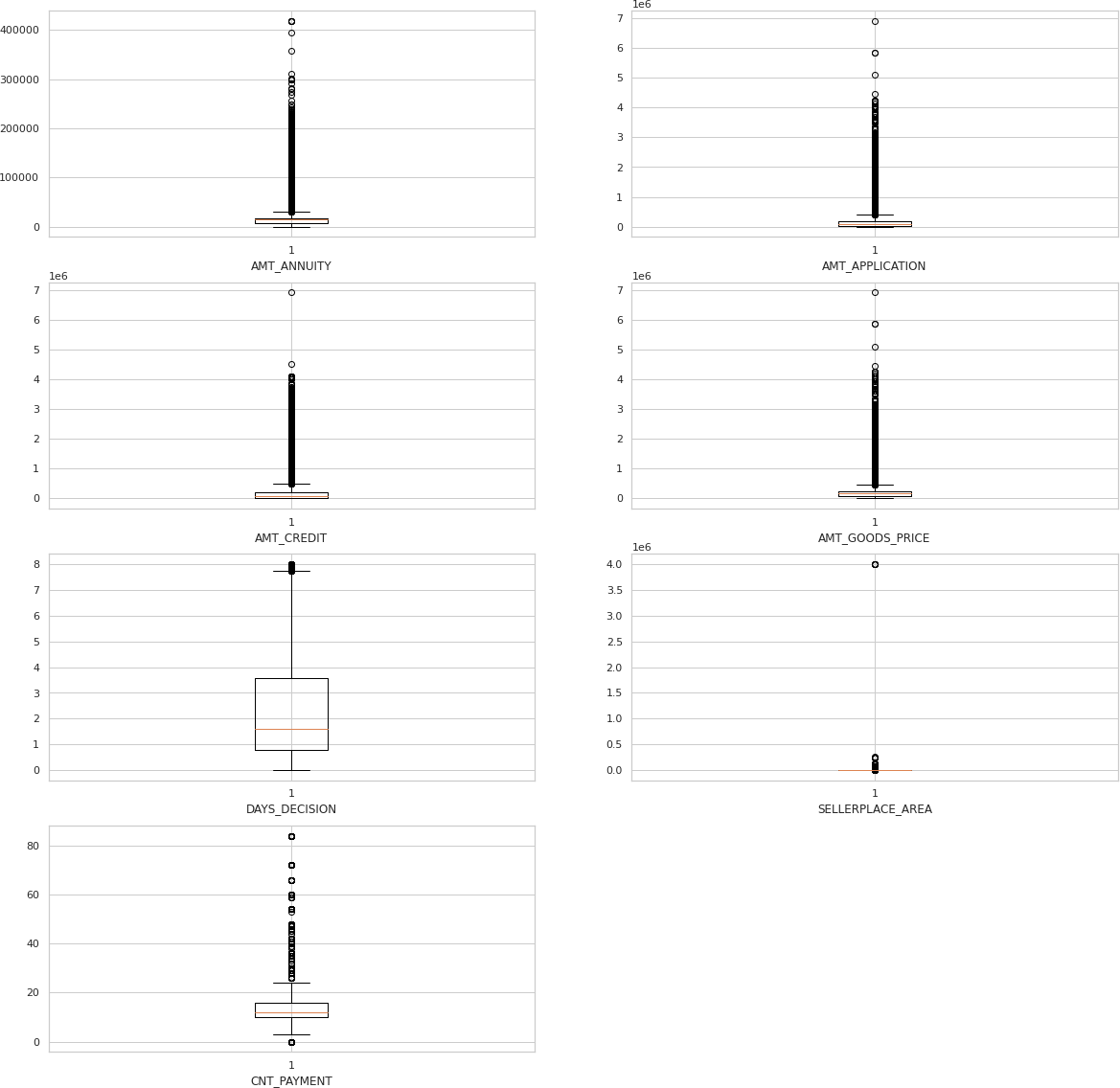
out\_check2 = 'AMT\_ANNUITY,AMT\_APPLICATION,AMT\_CREDIT,AMT\_GOODS\_PRICE,DAYS\_ DECISION,SELLERPLACE\_AREA,CNT\_PAYMENT'.split(',')

plt.figure(figsize=(20,20))

#df1.boxplot(column = out\_check, grid=False, rot=90, fontsize=15) for i in range(len(out\_check2)):

plt.subplot(4,2,i+1)

plt.boxplot(df[out\_check2[i]]) plt.xlabel(out\_check2[i])



Inference:

Each of the numerical column contain outliers.

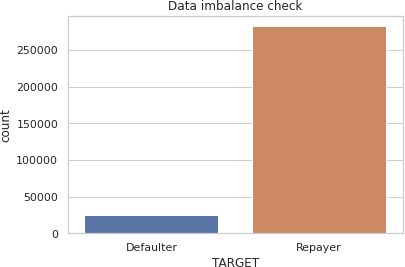
1. **Data Imbalance**

# before moving to the data imbalance rename the data values 0 to repayer and 1 to defaulter for better und

erstanding

df1["TARGET"] = df1["TARGET"].replace({1:"Defaulter",0:"Repayer"})

##### # Plot Target column



##### # percentage value count of defaulter and repayer

df1['TARGET'].value\_counts()\*100/len(df1) Repayer 91.927118

Defaulter 8.072882

Name: TARGET, dtype: float64

1. **Univariate Analysis for application\_data.csv**

##### # Function to write labels in the bar plot

def addlabels(x,y):

for i in range(len(x)):

plt.text(i, y[i]//2, y[i], ha = 'center')

# function for univariate analysis

# Cateriogical and numerical def univariate(df,data,target): col = df[data]

tar = df[target] types = col.dtypes

ex = col.value\_counts() if len(ex)>8:

plt.figure(figsize = (20,12)) sns.countplot(x = col, hue = tar) plt.xticks(rotation = 90)

plt.title(data,fontdict={'fontsize': 20}) elif len(ex)>4:

plt.figure(figsize = (15,8)) sns.countplot(x = col, hue = tar) plt.xticks(rotation = 45)

plt.title(data,fontdict={'fontsize': 20}) else:

plt.figure()

sns.countplot(x = col, hue = tar) plt.title(data,fontdict={'fontsize': 14}) plt.xlabel(data)

# Calculate defaulter % for the data

pf = df[[data,target]].value\_counts().reset\_index() percent= []

for i in pf[data].unique():

try:

percent.append(pf[(pf[data]==i)&(pf[target]=='Defaulter')][0].values[0]\*100/(pf[(pf[data]==i)&(pf[tar get]=='Defaulter')][0].values[0]+pf[(pf[data]==i)&(pf[target]=='Repayer')][0].values[0]))

except:

percent.append(0) if len(percent)>8:

plt.figure(figsize = (20,12))

sns.barplot(y = percent, x = pf[data].unique()) plt.xticks(rotation = 90)

plt.title('Defaulter % in '+data,fontdict={'fontsize': 20}) elif len(percent)>4:

plt.figure(figsize = (15,8))

sns.barplot(y = percent, x = pf[data].unique()) plt.xticks(rotation = 45)

plt.title('Defaulter % in '+data,fontdict={'fontsize': 20}) else:

plt.figure()

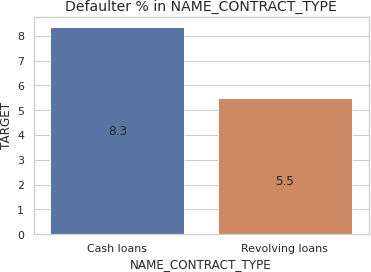
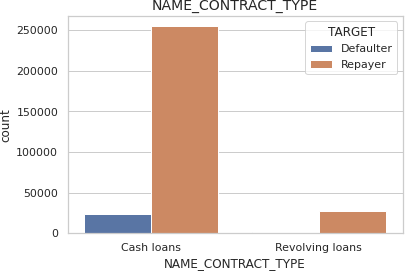
sns.barplot(y = percent, x = pf[data].unique()) plt.title('Defaulter % in '+data,fontdict={'fontsize': 14}) plt.xlabel(data)

plt.ylabel(target)

addlabels(x = pf[data].unique(),y=np.round(percent,1))

##### # Check for NAME\_CONTRACT\_TYPE based on loan repay status

univariate(df1,'NAME\_CONTRACT\_TYPE','TARGET')

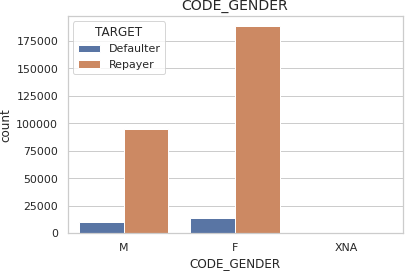


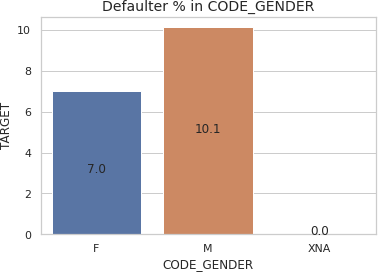
Inference:

* Revolving loans are only 8-12% compared to cash loans.
* Revolving loans have less defaulter%

##### # Check for CODE\_GENDER based on loan repay status

univariate(df1,'CODE\_GENDER','TARGET')



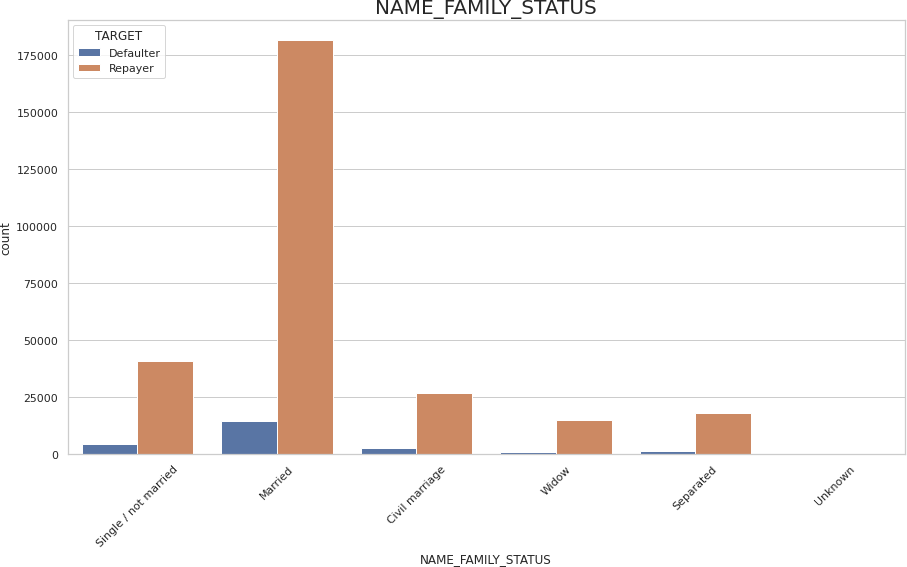


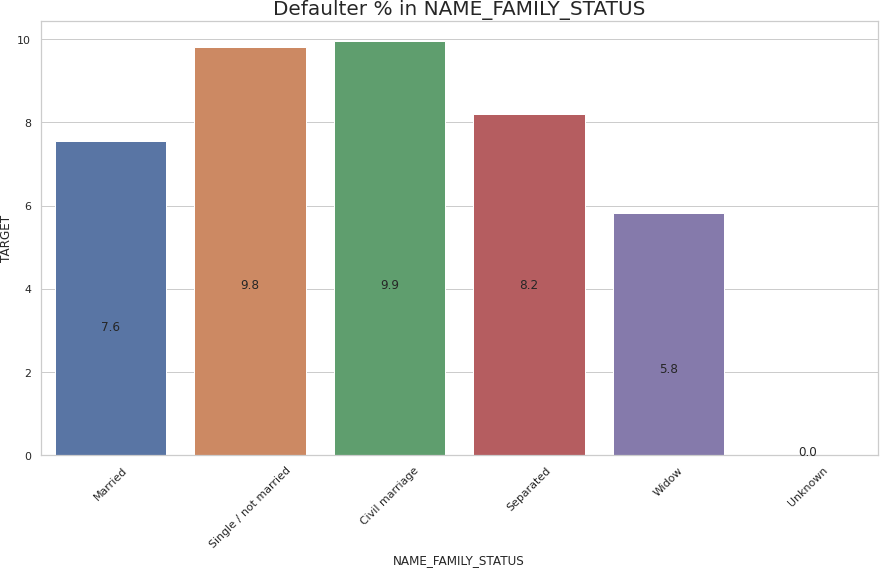
Inference:

* Females are most likely to take loans compared to males almost twice.
* Male have high defaulter than female.
* There are only few cases where people have not shared the gender but they paid all the loan.

##### # Check for NAME\_FAMILY\_STATUS based on loan repay status

univariate(df1,'NAME\_FAMILY\_STATUS','TARGET')



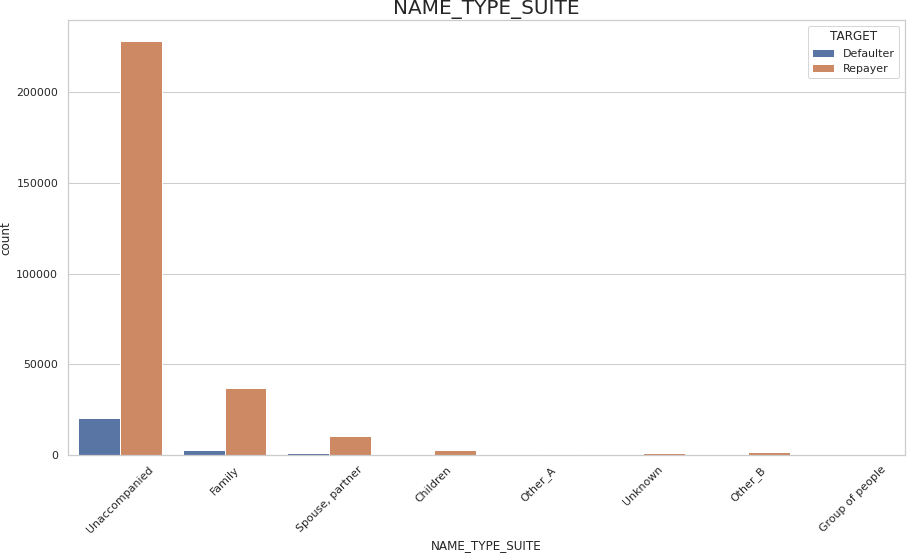


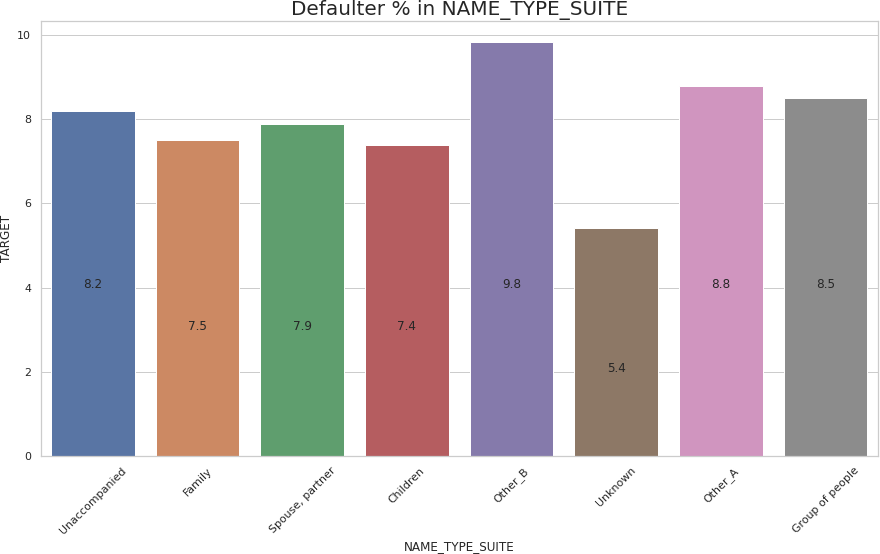
Inference:

* Married Family has taken more loans compared to all the category.
* In all the categories civil marriage have the highest default % followed by single / not married, separated, Married, and widow.
* Widow has the lowest defaulter value.

##### # Check for NAME\_TYPE\_SUITE based on loan repay status

univariate(df1,'NAME\_TYPE\_SUITE','TARGET')



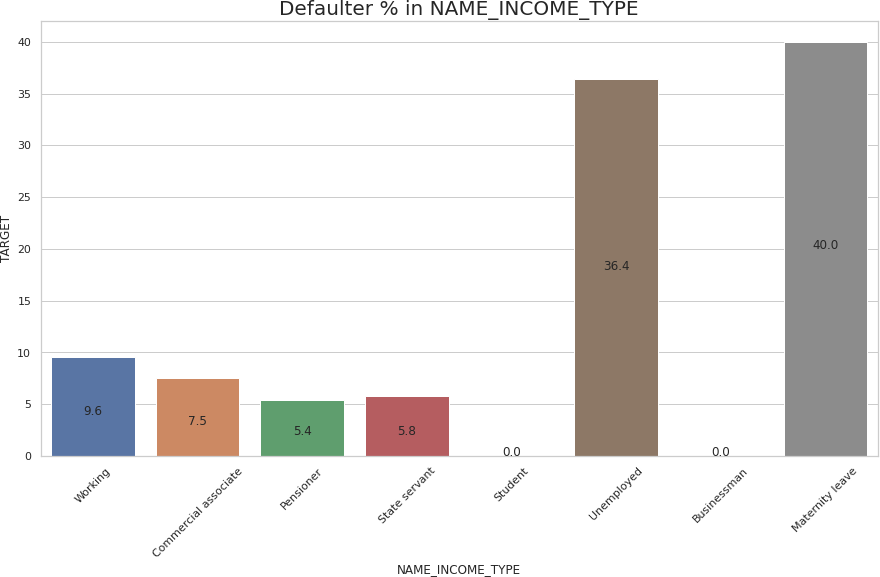
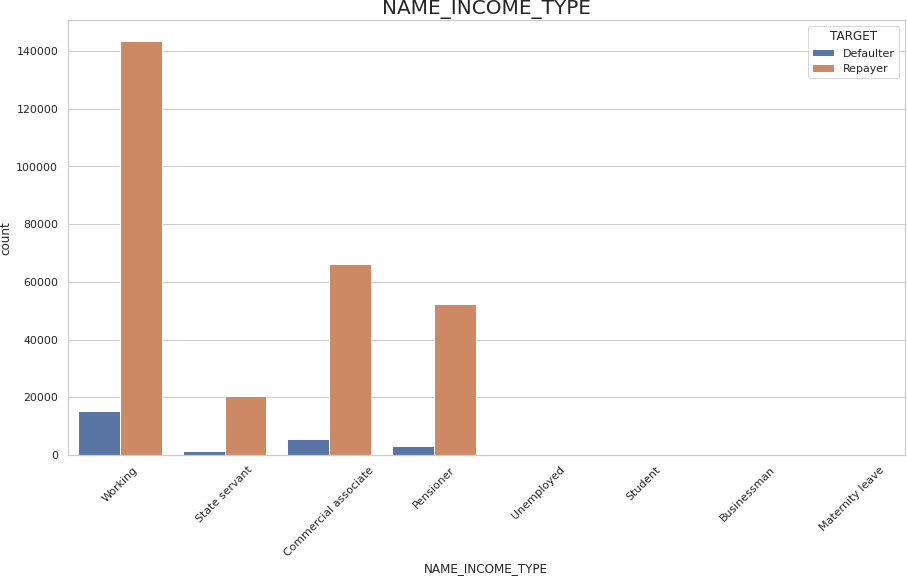


Inference:

* Accompanied category has the highest loans.
* The maximum and minimum defaulter% are Other\_B (9.8%) and Unknown (5.4%).

##### # Check for NAME\_INCOME\_TYPE based on loan repay status

univariate(df1,'NAME\_INCOME\_TYPE','TARGET')

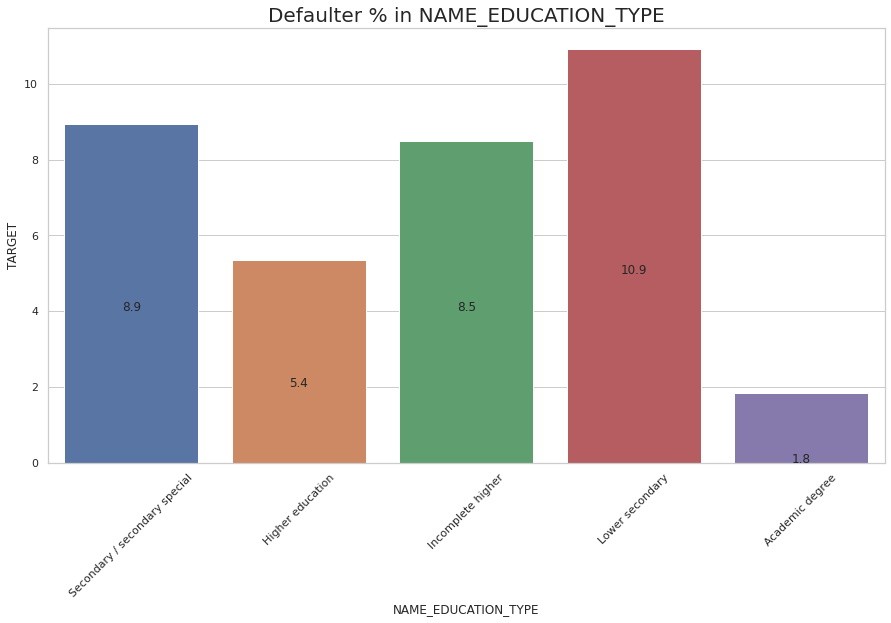


Inference:

* Working is more likely to take a loan, followed by commercial assistant, pensioner, etc.
* People on maternity leave is the riskiest category followed by unemployed, rest lies in less than 10%.
* Student and business are less in number but no records of defaulter hence it is the safest category to provide the loan.

##### # Check for NAME\_EDUCATION\_TYPE based on loan repay status

univariate(df1,'NAME\_EDUCATION\_TYPE','TARGET')

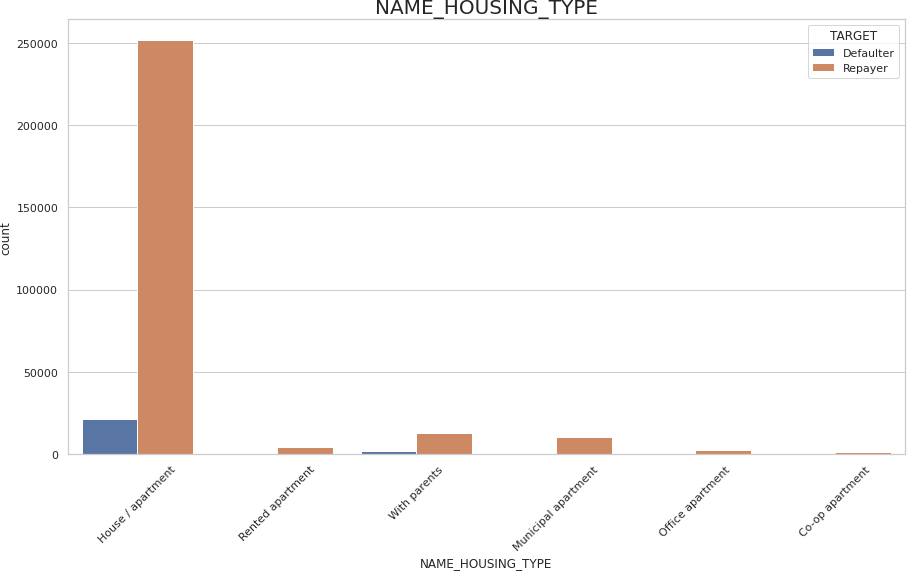


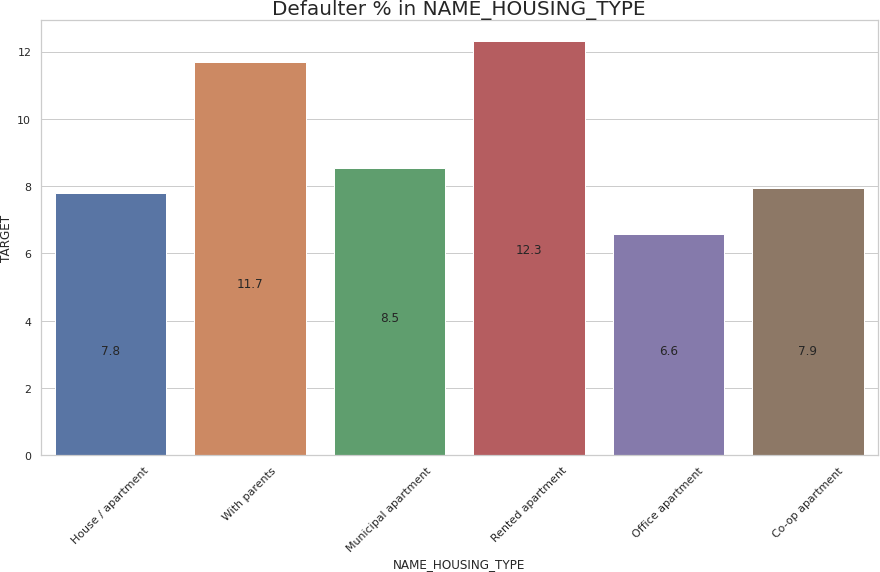
Inference:

* People with the secondary / secondary special category has the hights loan application.
* lower secondary has the highest defaulter%.
* Academic degree has the lowest defaulter% among all the categories.

##### # Check for NAME\_HOUSING\_TYPE based on loan repay status

univariate(df1,'NAME\_HOUSING\_TYPE','TARGET')



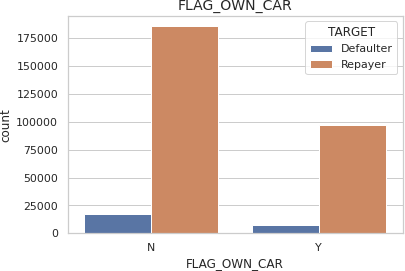


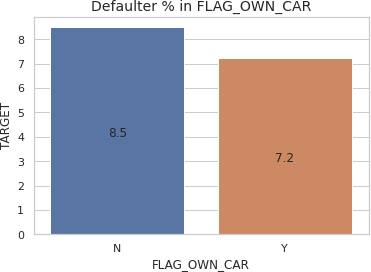
Inference:

* People with the house / apartment have the highest application.
* Rented apartment is the riskiest category among all.

##### # Check for FLAG\_OWN\_CAR based on loan repay status

univariate(df1,'FLAG\_OWN\_CAR','TARGET')



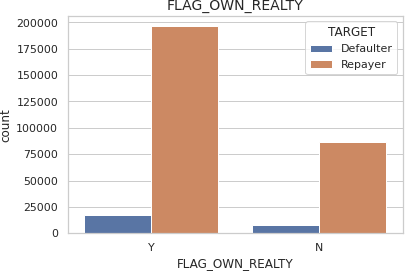


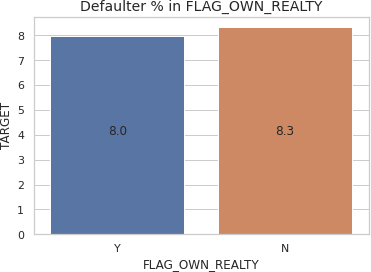
Inference:

* People who do not own car have high loan applications as well as high defaulter%.

##### # Check for FLAG\_OWN REALTY type based on loan repay status

univariate(df1,'FLAG\_OWN\_REALTY','TARGET')



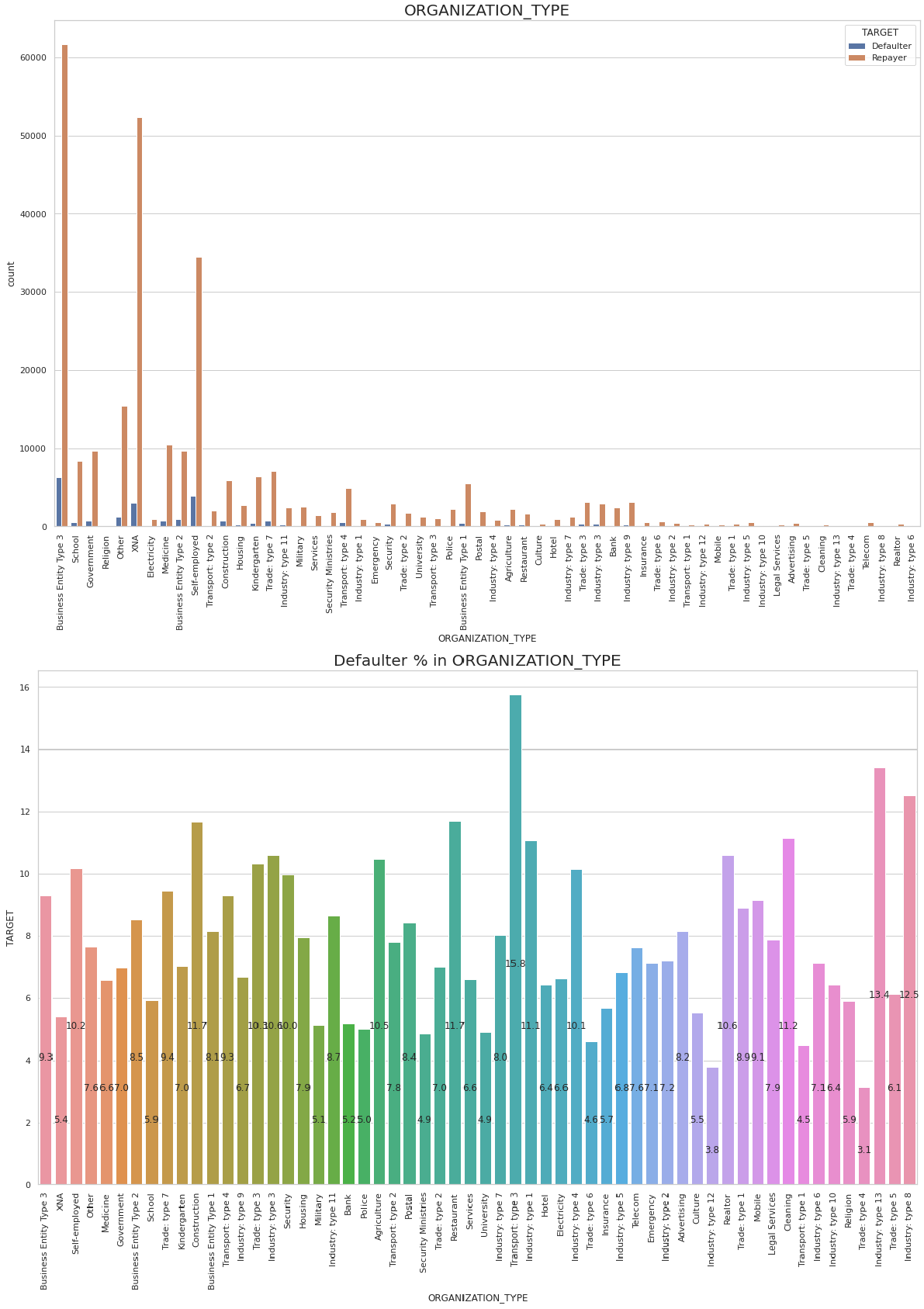


Inference:

* people who own realty have high loan application and low defaulter% than who does not own realty.

##### # Check for ORGANIZATION\_TYPE based on loan repay status

univariate(df1,'ORGANIZATION\_TYPE','TARGET')



Inference:

* Business entity type 3, XNA and self employed have the highest loan application others are less than 15%.
* riskiest category is transport: type 3 (15.8%)

##### # Convert AMT\_INCOME\_TOTAL to bins

incomebins=[0,25000,50000,75000,100000,125000,150000,175000,200000,225000,250000,275000,3000 00,325000,350000,375000,400000,425000,450000,475000,500000,10000000000]

incomeslots = ['0-25000','25000-50000','50000-75000','75000,100000','100000-125000', '125000-

150000', '150000-175000','175000-200000','200000-225000','225000-250000','250000-275000','275000-

300000','300000-325000','325000-350000','350000-375000','375000-400000','400000-425000','425000-

450000','450000-475000','475000-500000','500000 and above']

df1['AMT\_INCOME\_TOTAL\_RANGE']=pd.cut(df1['AMT\_INCOME\_TOTAL'],bins=incomebins,labels

=incomeslots)

##### # convert AMT\_CREDIT to bins

creditbins = [0,150000,200000,250000,300000,350000,400000,450000,500000,550000,600000,650000,70

0000,750000,800000,850000,900000,1000000000]

creditslots = ['0-150000', '150000-200000','200000-250000', '250000-300000', '300000-350000', '350000-

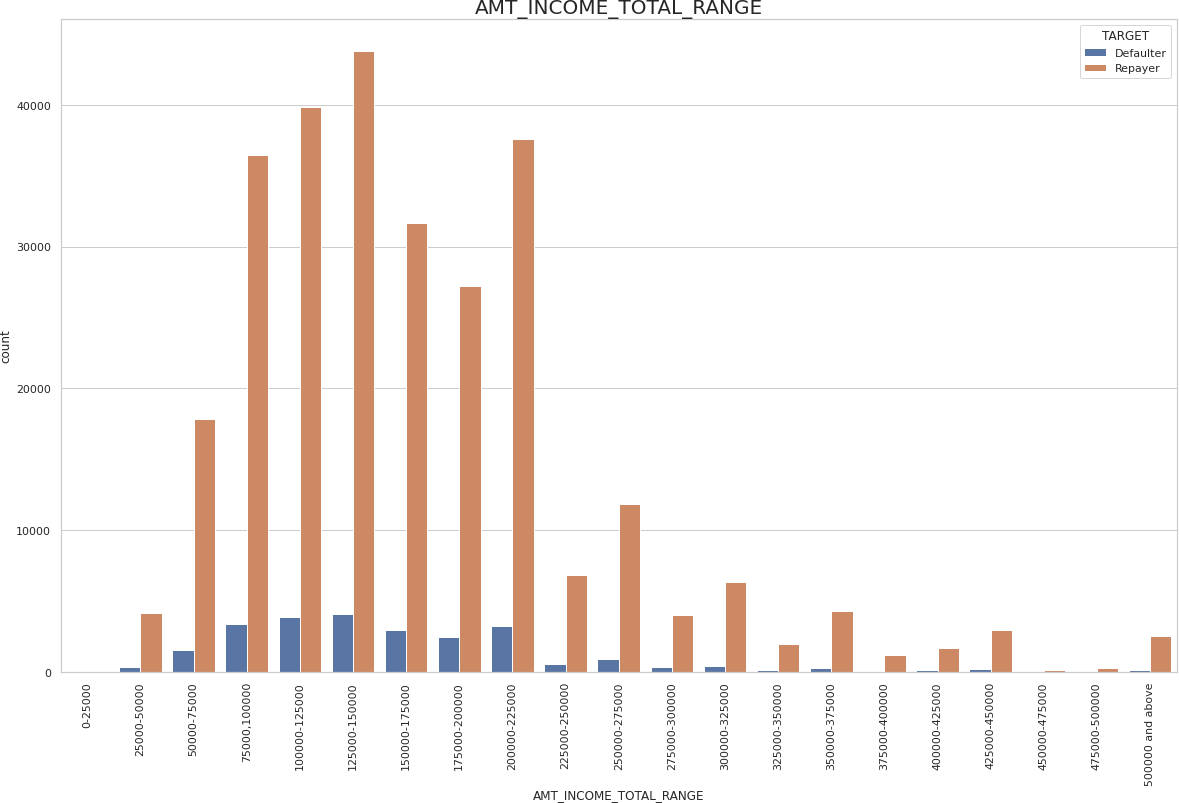
400000','400000-450000','450000-500000','500000-550000','550000-600000','600000-650000','650000-

700000','700000-750000','750000-800000','800000-850000','850000-900000','900000 and above']

df1['AMT\_CREDIT\_RANGE'] = pd.cut(df1.AMT\_CREDIT,bins=creditbins,labels=creditslots)

##### # univariate analysis for AMT\_INCOME\_TOTAL\_RANGE

univariate(df1,'AMT\_INCOME\_TOTAL\_RANGE','TARGET')





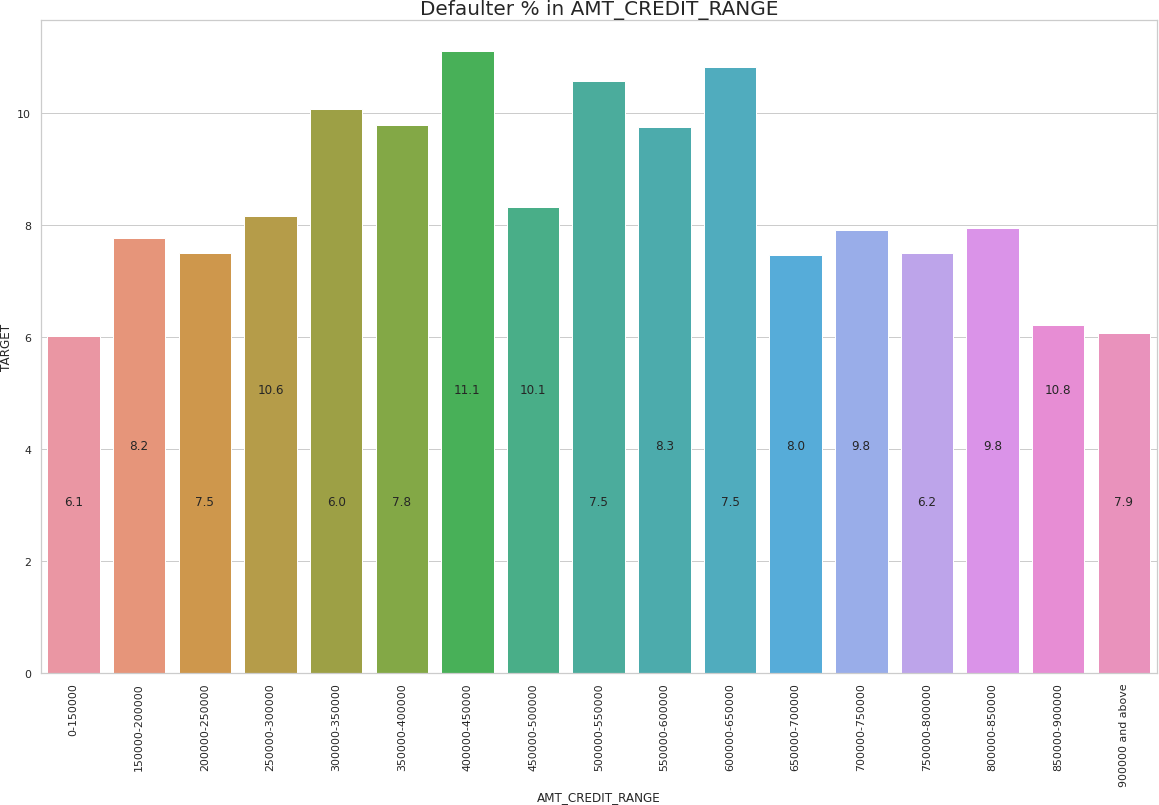
Inference:

* Majority of the people who apply for the loan lies between 75000 to 225000.
* people with low salary have higher default rate.
* exception to the above point salary from 475000 to 50000 have the higher default rate.



univariate(df1,'AMT\_CREDIT\_RANGE','TARGET')

**# Univariate analysis for AMT\_CREDIT**



Inference:

* Maximum loans are given in the range of 90000 and above.
* credit loan 30000 to 65000 have the higher default rate.

1. **Bivariate analysis for application\_data.csv**

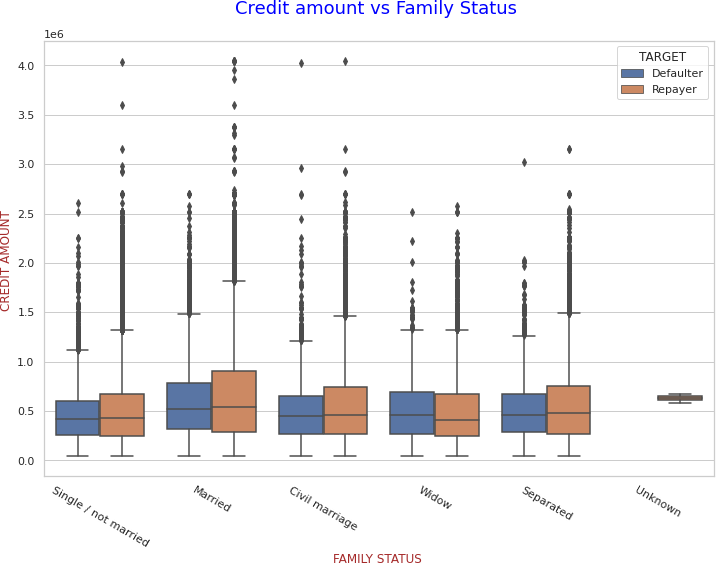
##### #Checking for Credit amount provided to the customers based on their Family type and plotted according TO target

plt.figure(figsize=(12,8)) scale\_factor=5

sns.boxplot(data=df1,x=df1.NAME\_FAMILY\_STATUS,y=df1.AMT\_CREDIT,hue=df1.TARGET) plt.xticks(rotation=-30)

plt.xlabel("FAMILY STATUS ", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'}) plt.ylabel("CREDIT AMOUNT ", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'}) plt.title('Credit amount vs Family Status \n',fontdict={'fontsize': 18, 'fontweight' : 10, 'color' : 'Blue'})

plt.show()



Inference:

* Civil Marriage has the highest defaulter credit amount.
* The credit amount for the repayers is high except in the case of civil marriage.

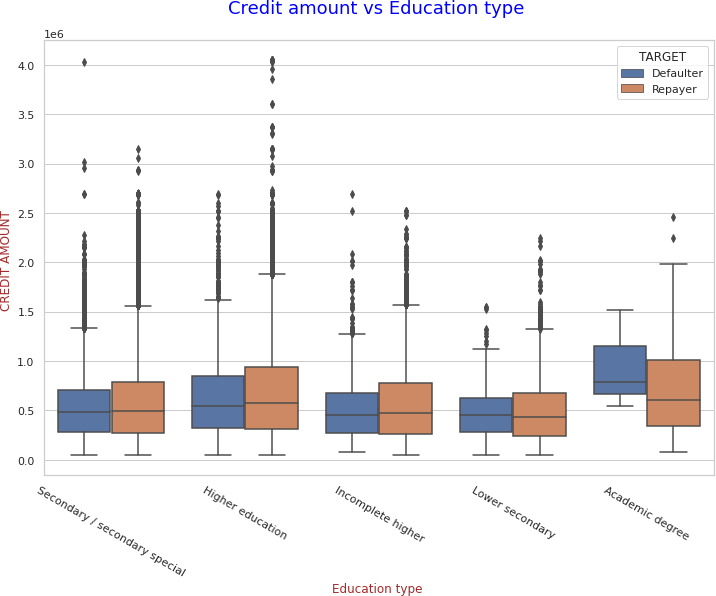
##### #Checking for Credit amount provided to the customers based on their Education type and plotted according TO target

plt.figure(figsize=(12,8)) scale\_factor=5

sns.boxplot(data=df1,x=df1.NAME\_EDUCATION\_TYPE,y=df1.AMT\_CREDIT,hue=df1.TARGET) plt.xticks(rotation=-30)

plt.xlabel("Education type ", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'}) plt.ylabel("CREDIT AMOUNT ", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'}) plt.title('Credit amount vs Education type \n',fontdict={'fontsize': 18, 'fontweight' : 10, 'color' : 'Blue'})

plt.show()



Inference:

* Higher education people are high credit seekers.
* Secondary / secondary special has high defaulter credit amount.

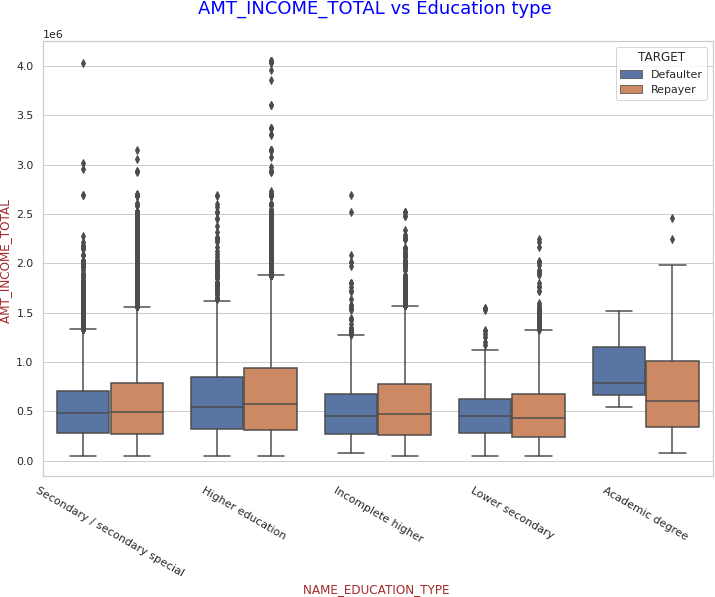
##### #Checking for AMT\_INCOME\_TOTAL provided to the customers based on their educ ation type and plotted according to target

plt.figure(figsize=(12,8)) scale\_factor=5

sns.boxplot(data=df1,x=df1.NAME\_EDUCATION\_TYPE,y=df1.AMT\_CREDIT,hue=df1.TARGET) plt.xticks(rotation=-30)

plt.xlabel("NAME\_EDUCATION\_TYPE ", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'}) plt.ylabel("AMT\_INCOME\_TOTAL ", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'}) plt.title('AMT\_INCOME\_TOTAL vs Education type \n',fontdict={'fontsize': 18, 'fontweight' : 10, 'color' : ' Blue'})

plt.show()



Inference:

* higher education is earning more than any education type.
* secondary / secondary special has the higher defaulter percentage.

##### # Pair plots for the numeric column

amount = df1[['AMT\_INCOME\_TOTAL','AMT\_CREDIT','AMT\_ANNUITY','AMT\_GOODS\_PRICE','T ARGET']]

sns.pairplot(amount,hue = 'TARGET')



Inference:

* 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. It forms a linear relation.
* There are very less defaulters for AMT\_CREDIT >20 Lakhs

1. **Correlation check for the defaulter in application\_data.csv**

##### # Divide the data based on the target column

defaulter\_df = df1[df1['TARGET']==1]

##### # Correlation check

corr2 = defaulter\_df[['NAME\_CONTRACT\_TYPE', 'CODE\_GENDER',

'FLAG\_OWN\_CAR', 'FLAG\_OWN\_REALTY', 'CNT\_CHILDREN', 'AMT\_INCOME\_TOTAL', 'AMT\_CREDIT', 'AMT\_ANNUITY', 'AMT\_GOODS\_PRICE', 'NAME\_TYPE\_SUITE', 'NAME\_INCOME\_TYPE', 'NAME\_EDUCATION\_TYPE', 'NAME\_FAMILY\_STATUS',

'NAME\_HOUSING\_TYPE', 'REGION\_POPULATION\_RELATIVE', 'AGE',

Top 10 correlation

Var1

Var2 Correlation

1. AMT\_GOODS\_PRICE AMT\_CREDIT 0.982566
2. REGION\_RATING\_CLIENT\_W\_CITY REGION\_RATING\_CLIENT 0.956637

2

3

4

5

6

7

8

9

CNT\_FAM\_MEMBERS AMT\_ANNUITY

AMT\_GOODS\_PRICE

CNT\_CHILDREN 0.885484

AMT\_CREDIT

0.752195

AMT\_ANNUITY

FLAG\_PHONE

FLAG\_WORK\_PHONE

0.752022

0.311035

YEARS\_REGISTRATION YEARS\_ID\_PUBLISH

AGE

0.289114

AGE 0.252863

AGE AGE

AMT\_GOODS\_PRICE

0.135754

AMT\_CREDIT

0.135316

'YEARS\_REGISTRATION', 'YEARS\_ID\_PUBLISH', 'FLAG\_WORK\_PHONE', 'FLAG\_PHONE', 'OCCUPATION\_TYPE', 'CNT\_FAM\_MEMBERS',

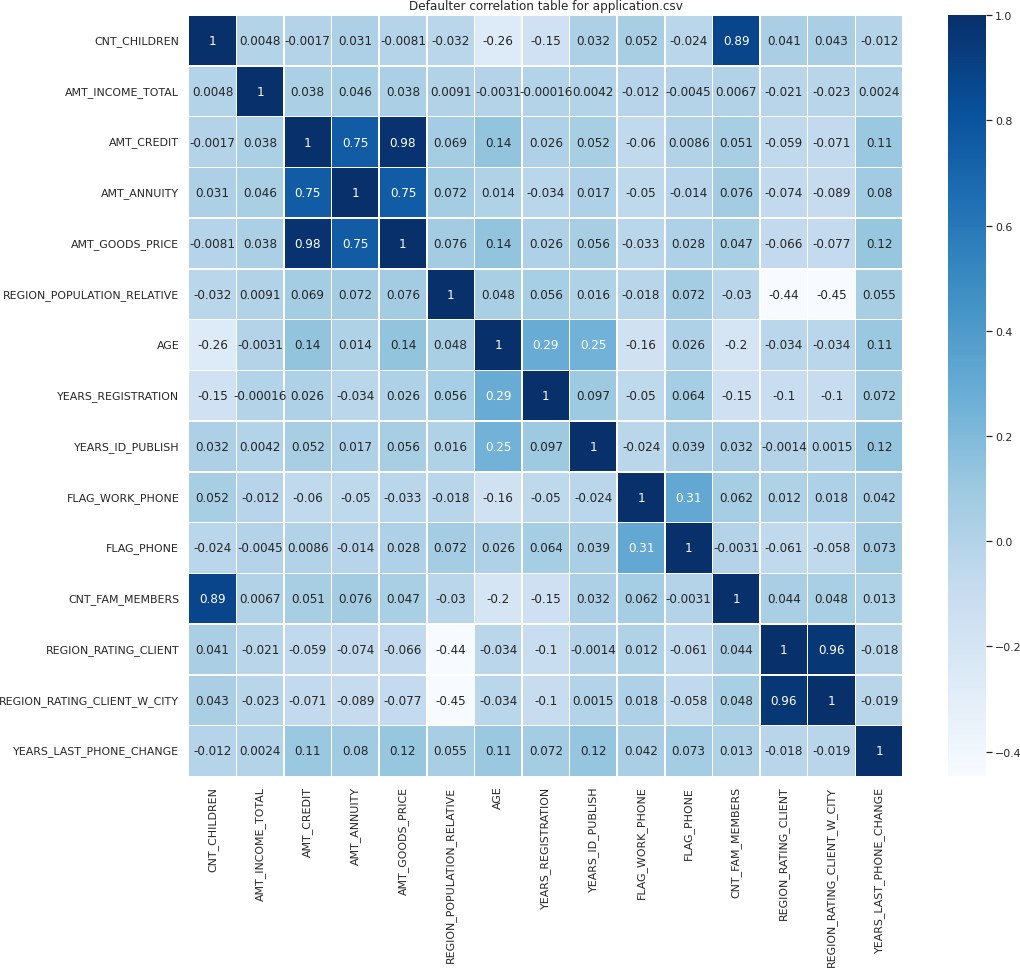
'REGION\_RATING\_CLIENT', 'REGION\_RATING\_CLIENT\_W\_CITY','ORGANIZATION\_TYPE', 'YEARS\_LAST\_PHONE\_CHANGE']].corr()

plt.figure(figsize=(16,14)) sns.heatmap(corr2,cmap='Blues',annot=True,linewidth=0.5) plt.title('Defaulter correlation table for application.csv')

corr2 = corr2.where(np.triu(np.ones(corr2.shape), k=1).astype(bool)) corr\_mat = corr2.unstack().sort\_values(ascending=False).reset\_index()

corr\_mat.rename(columns={'level\_0':'Var1','level\_1':'Var2',0:'Correlation'},inplace=True )

print(corr\_mat.head(10))



1. **Univariate analysis of merged data frame**

##### # Merge the data set on SK\_ID\_CURR

mergedf = df1.merge(df, on='SK\_ID\_CURR')

##### # Renaming the columns as convention

mergedf = mergedf.rename({'NAME\_CONTRACT\_TYPE\_x' : 'NAME\_CONTRACT\_TYPE','AMT\_CR EDIT\_x':'AMT\_CREDIT','AMT\_ANNUITY\_x':'AMT\_ANNUITY',

'WEEKDAY\_APPR\_PROCESS\_START\_x' : 'WEEKDAY\_APPR\_PROCESS\_START',' AMT\_GOODS\_PRICE\_x':'AMT\_GOODS\_PRICE',

'HOUR\_APPR\_PROCESS\_START\_x':'HOUR\_APPR\_PROCESS\_START','NAME\_CON TRACT\_TYPE\_y':'NAME\_CONTRACT\_TYPE\_PREV',

'AMT\_CREDIT\_y':'AMT\_CREDIT\_PREV','AMT\_ANNUITY\_y':'AMT\_ANNUITY\_PRE V','AMT\_GOODS\_PRICE\_y':'AMT\_GOODS\_PRICE\_PREV',

'WEEKDAY\_APPR\_PROCESS\_START\_y':'WEEKDAY\_APPR\_PROCESS\_START\_PR

EV',

'HOUR\_APPR\_PROCESS\_START\_y':'HOUR\_APPR\_PROCESS\_START\_PREV'}, axis

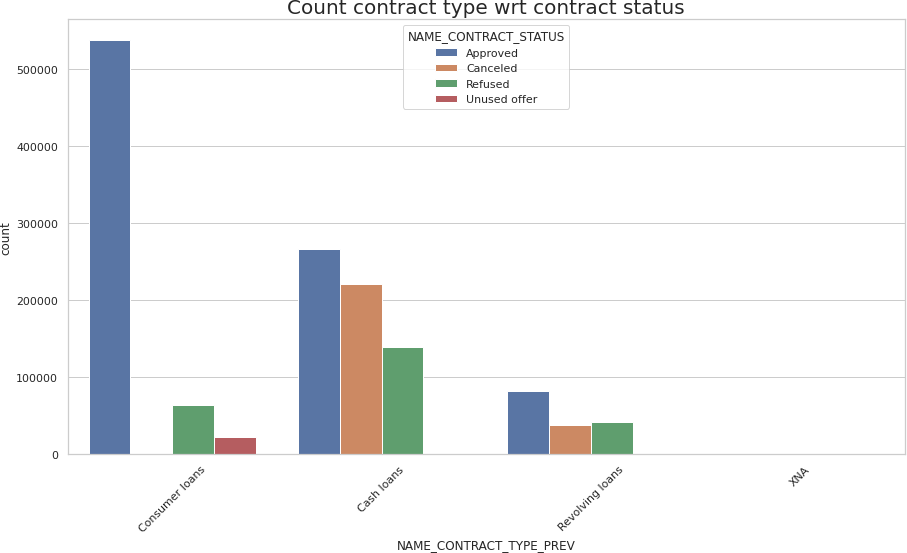
=1)

plt.figure(figsize = (15,8))

sns.countplot(x = mergedf['NAME\_CONTRACT\_TYPE\_PREV'], hue = mergedf['NAME\_CONTRACT\_ STATUS'])

plt.xticks(rotation = 45)

plt.title('Count contract type wrt contract status',fontdict={'fontsize': 20})



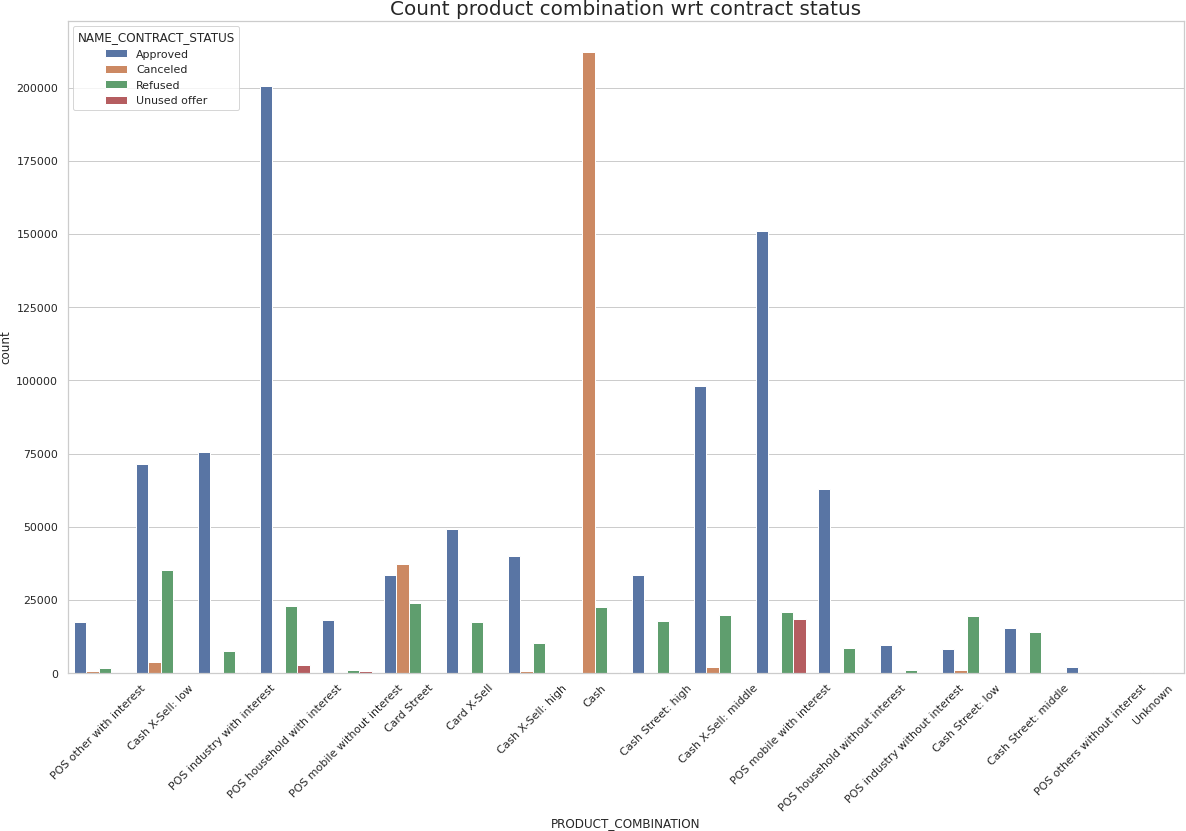
Inference:

* Consumer loans are highly applied and highly approved.
* There are only few cancelled loans in consumer loans.
* Revolving loans are less applied.
* More than 70% of cash loans are cancelled.

##### # univariate analysis of PRODUCT\_COMBINATION with hue NAME\_CONTRACT\_STATUS

plt.figure(figsize = (20,12)) sns.countplot(mergedf['PRODUCT\_COMBINATION'],hue=mergedf['NAME\_CONTRACT\_STATUS']) plt.xticks(rotation = 45)

plt.title('Count product combination wrt contract status',fontdict={'fontsize': 20})



Inference:

* POS industry with interest has the highest approved loans.
* Cash loans has the highest cancelled loans.

##### # Univariate analysis of CODE\_GENDER with hue NAME\_CONTRACT\_STATUS

plt.figure(figsize = (15,8)) sns.countplot(mergedf['CODE\_GENDER'],hue=mergedf['NAME\_CONTRACT\_STATUS']) plt.xticks(rotation = 45)

plt.title('Count gender wrt contract status',fontdict={'fontsize': 20})



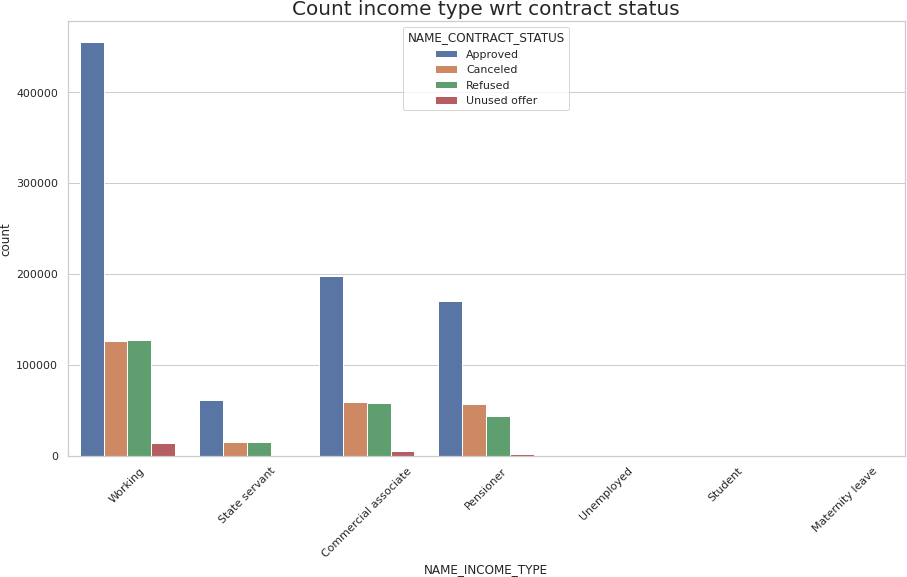
Inference:

* Female loans are Approved more than male.
* Female loans have high unused offers than male.

##### # Univariate analysis of NAME\_INCOME\_TYPE with hue NAME\_CONTRACT\_STATUS

plt.figure(figsize = (15,8)) sns.countplot(mergedf['NAME\_INCOME\_TYPE'],hue=mergedf['NAME\_CONTRACT\_STATUS']) plt.xticks(rotation = 45)

plt.title('Count income type wrt contract status',fontdict={'fontsize': 20})



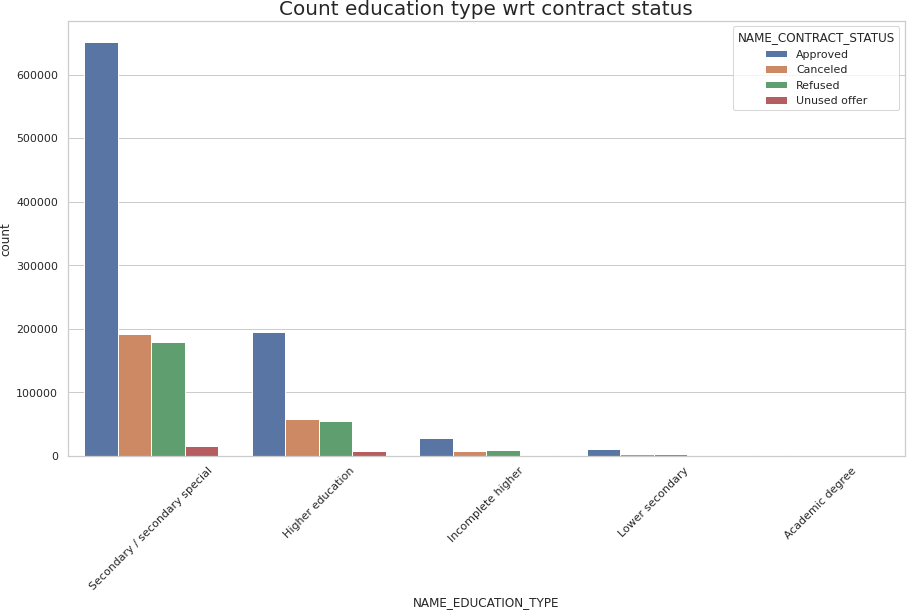
Inference:

* loans for working people are approved more.
* unemployed, student and maternity leave have very few records.

##### # Univariate analysis of NAME\_EDUCATION\_TYPE with hue NAME\_CONTRACT\_STATUS

plt.figure(figsize = (15,8)) sns.countplot(mergedf['NAME\_EDUCATION\_TYPE'],hue=mergedf['NAME\_CONTRACT\_STATUS']) plt.xticks(rotation = 45)

plt.title('Count education type wrt contract status',fontdict={'fontsize': 20})



Inference:

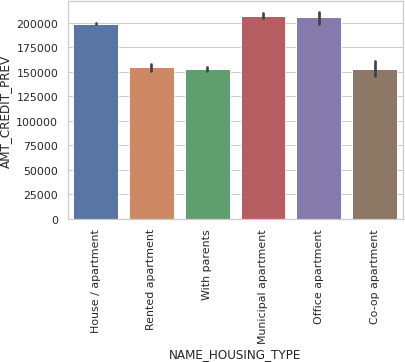
* secondary education has the highest loan approved.
* Academic degree has very few records

1. **Bivariate analysis of merged data frame**

##### # Bar plot between NAME\_HOUSING\_TYPE and AMT\_CREDIT\_PREV

sns.barplot(data=mergedf,x = 'NAME\_HOUSING\_TYPE',y = 'AMT\_CREDIT\_PREV' )

plt.xticks(rotation=90)



Inference:

* Municipal apartment and credit apartment have the highest amount credit.

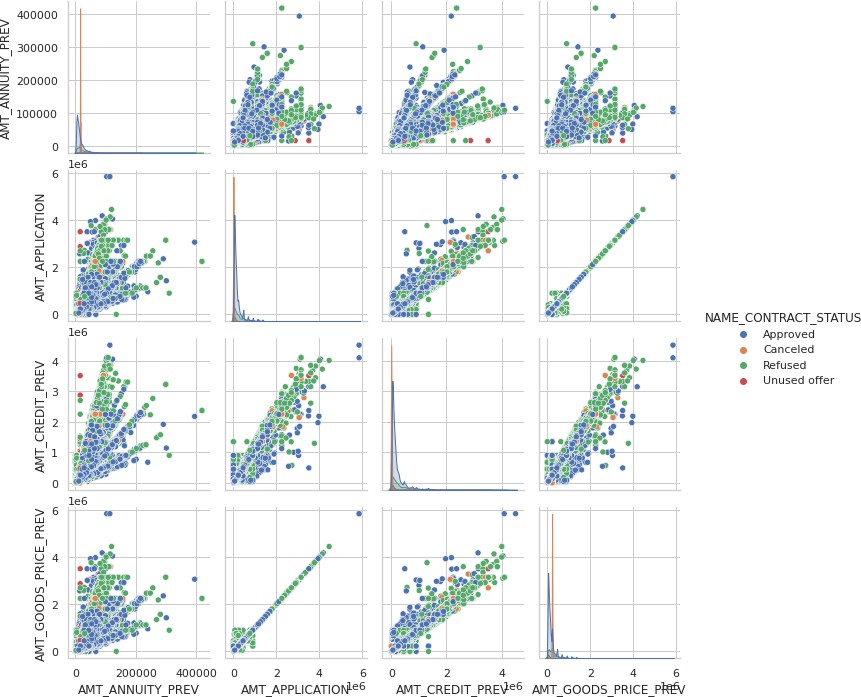
1. **Correlation check for previous\_application.csv**

##### # Pair plots for the numeric column

plt.figure(figsize=(20,14))

amount = mergedf[['AMT\_ANNUITY\_PREV', 'AMT\_APPLICATION', 'AMT\_CREDIT\_PREV', 'AMT\_GOODS\_PRICE\_PREV','NAME\_CONTRACT\_STATUS']]

sns.pairplot(amount,hue = 'NAME\_CONTRACT\_STATUS')



Inference:

* AMT\_GOODS\_PRICE\_PREV and AMT\_APPLICATION shows too high linear relation.

##### # Extracting columns for the correlation check

corr4 = mergedf[['TARGET', 'NAME\_CONTRACT\_TYPE', 'CODE\_GENDER', 'FLAG\_OWN\_CAR', 'FLAG\_OWN\_REALTY', 'CNT\_CHILDREN', 'AMT\_INCOME\_TOTAL', 'AMT\_CREDIT', 'AMT\_ANNUITY', 'AMT\_GOODS\_PRICE', 'NAME\_TYPE\_SUITE', 'NAME\_INCOME\_TYPE', 'NAME\_EDUCATION\_TYPE', 'NAME\_FAMILY\_STATUS', 'NAME\_HOUSING\_TYPE', 'REGION\_POPULATION\_RELATIVE', 'AGE', 'YEARS\_REGISTRATION', 'YEARS\_ID\_PUBLISH', 'FLAG\_WORK\_PHONE', 'FLAG\_PHONE', 'OCCUPATION\_TYPE', 'CNT\_FAM\_MEMBERS', 'REGION\_RATING\_CLIENT', 'REGION\_RATING\_CLIENT\_W\_CITY', 'REG\_REGION\_NOT\_LIVE\_REGION', 'REG\_REGION\_NOT\_WORK\_REGION', 'LIVE\_REGION\_NOT\_WORK\_REGION', 'REG\_CITY\_NOT\_LIVE\_CITY', 'REG\_CITY\_NOT\_WORK\_CITY', 'LIVE\_CITY\_NOT\_WORK\_CITY', 'ORGANIZATION\_TYPE', 'OBS\_30\_CNT\_SOCIAL\_CIRCLE', 'DEF\_30\_CNT\_SOCIAL\_CIRCLE', 'OBS\_60\_CNT\_SOCIAL\_CIRCLE', 'DEF\_60\_CNT\_SOCIAL\_CIRCLE', 'YEARS\_LAST\_PHONE\_CHANGE', 'AMT\_REQ\_CREDIT\_BUREAU\_HOUR', 'AMT\_REQ\_CREDIT\_BUREAU\_DAY',

'AMT\_REQ\_CREDIT\_BUREAU\_WEEK', 'AMT\_REQ\_CREDIT\_BUREAU\_MON',

Var1

Var2

0

1

2

3

OBS\_60\_CNT\_SOCIAL\_CIRCLE AMT\_GOODS\_PRICE AMT\_CREDIT\_PREV AMT\_GOODS\_PRICE\_PREV

Correlation

OBS\_30\_CNT\_SOCIAL\_CIRCLE 0.998561

AMT\_CREDIT

0.985959

AMT\_APPLICATION

0.975683

AMT\_APPLICATION

0.945759

4 REGION\_RATING\_CLIENT\_W\_CITY REGION\_RATING\_CLIENT 0.945596

5

6

AMT\_GOODS\_PRICE\_PREV CNT\_FAM\_MEMBERS

AMT\_CREDIT\_PREV 0.939147

CNT\_CHILDREN

0.879224

1. LIVE\_REGION\_NOT\_WORK\_REGION REG\_REGION\_NOT\_WORK\_REGION 0.875505
2. DEF\_60\_CNT\_SOCIAL\_CIRCLE DEF\_30\_CNT\_SOCIAL\_CIRCLE 0.862698
3. LIVE\_CITY\_NOT\_WORK\_CITY REG\_CITY\_NOT\_WORK\_CITY 0.831422

'AMT\_REQ\_CREDIT\_BUREAU\_QRT', 'AMT\_REQ\_CREDIT\_BUREAU\_YEAR', 'AMT\_INCOME\_TOTAL\_RANGE', 'AMT\_CREDIT\_RANGE', 'NAME\_CONTRACT\_TYPE\_PREV', 'AMT\_ANNUITY\_PREV', 'AMT\_APPLICATION', 'AMT\_CREDIT\_PREV', 'AMT\_GOODS\_PRICE\_PREV', 'NAME\_CASH\_LOAN\_PURPOSE', 'NAME\_CONTRACT\_STATUS', 'DAYS\_DECISION', 'NAME\_PAYMENT\_TYPE', 'CODE\_REJECT\_REASON', 'NAME\_CLIENT\_TYPE', 'NAME\_GOODS\_CATEGORY', 'NAME\_PORTFOLIO', 'NAME\_PRODUCT\_TYPE', 'CHANNEL\_TYPE', 'SELLERPLACE\_AREA', 'NAME\_SELLER\_INDUSTRY', 'CNT\_PAYMENT',

'NAME\_YIELD\_GROUP', 'PRODUCT\_COMBINATION']].corr()

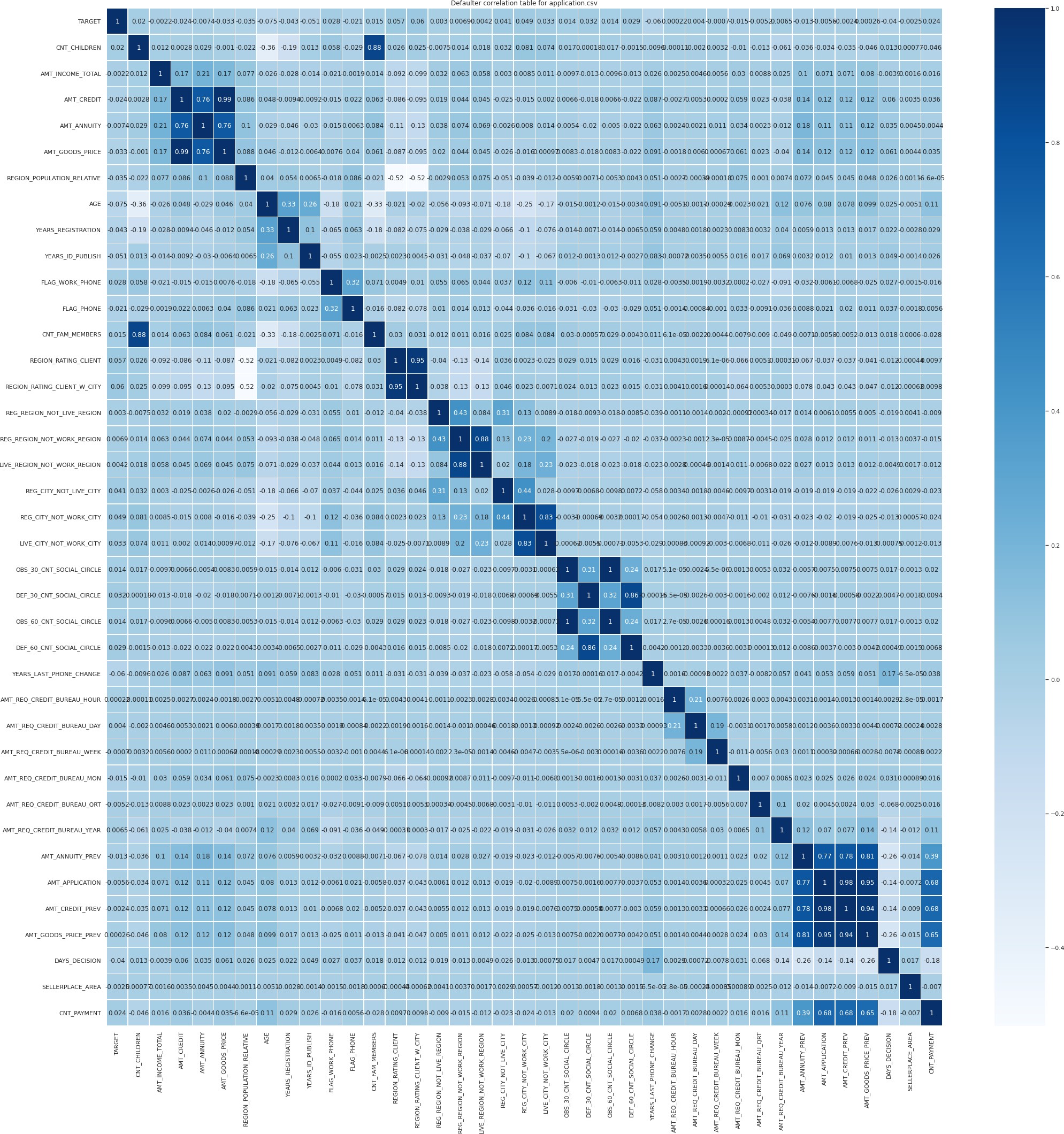
##### # Plot correlation map

plt.figure(figsize=(35,35)) sns.heatmap(corr4,cmap='Blues',annot=True,linewidth=0.5) plt.title('Defaulter correlation table for application.csv')

corr5 = corr4.where(np.triu(np.ones(corr4.shape), k=1).astype(bool)) corr\_mat = corr5.unstack().sort\_values(ascending=False).reset\_index()

corr\_mat.rename(columns={'level\_0':'Var1','level\_1':'Var2',0:'Correlation'},inplace=True )

print(corr\_mat.head(10))



## Conclusion:

* The data is highly imbalance with almost 92% repayor and 8% defaulter.
* It is found that revolving loans has less defaulter (5.5%) than cash loans (8.2%).
* Unemployed and Maternity leave, income types have the highest defaulter 36.5% and 40% respectively. Loans giving to this category should be avoided.
* Academic degree in education type has the minimum defaulter rate. Loan distribution to this category can be increased.
* Person who owns car have the less defaulter rate than who does not own car.
* Banks should focus more on contract type ‘Student’, ‘pensioner’ and ‘Businessman’ with housing ‘type other than ‘Co-op apartment’ and 'office apartment' for successful payments.
* OCCUPATION\_TYPE: Avoid Low-skill Laborers, Drivers and Waiters/barmen staff, Security staff, Laborers and Cooking staff as their default rate is huge.
* AMT\_GOODS\_PRICE: When the credit amount goes beyond 3lakhs, there is an increase in defaulters.

## Challenges:

* The Dataset contains 122 features in application\_data.csv and 32 features in previous\_application.csv which makes it complicated to find the relation between the dataset.
* The processing time while making pair plots and correlation plots is quite high because of huge dataset.

# Module 7: Impact of Car Features

## Project Description:

In this project we needed to derive various insights into how various car features affect the price and profitability of the car. Various columns show different features and the relationship between the car's features, market category, and pricing, these are some features and categories which are most popular among consumers and most profitable for the manufacturer. We needed to use different analysis techniques like regression analysis, market segmentation, correlation analysis, etc to get insights about the car and how the car performs in a given market. Using Pivot tables for insights and then for data visualization using different charts, then making an interactive dashboard using slicers and filters.

## Approach:

The approach was to first study and understand each column of the given dataset. Then select specific columns to find relationships between various parameters and derive insights from them. After getting the insights, for data visualization then they are converted into charts from tables. So, the insights are represented in a better way.

## Tech-Stack Used:

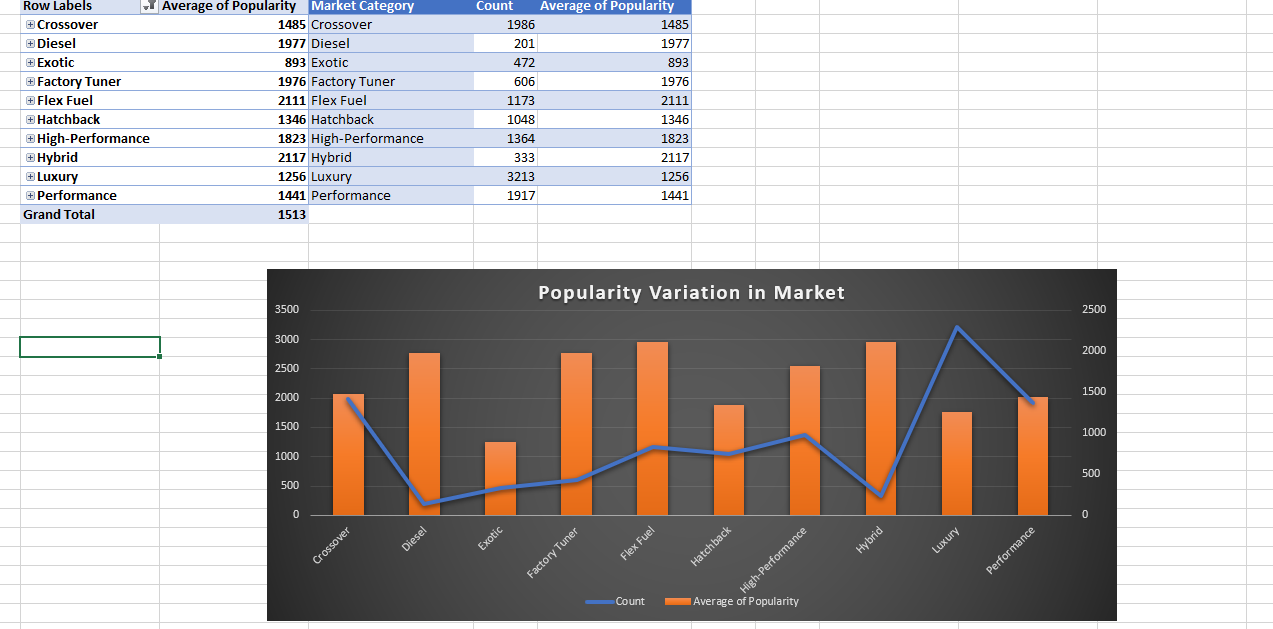
For this assignment I have used Microsoft Excel (2019) and Microsoft PowerPoint (2019)

## Insights:

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

**Task 1.A:** Create a pivot table that shows the number of car models in each market category and their corresponding popularity scores.

**Task 1.B:** Create a combo chart that visualizes the relationship between market category and popularity.

****

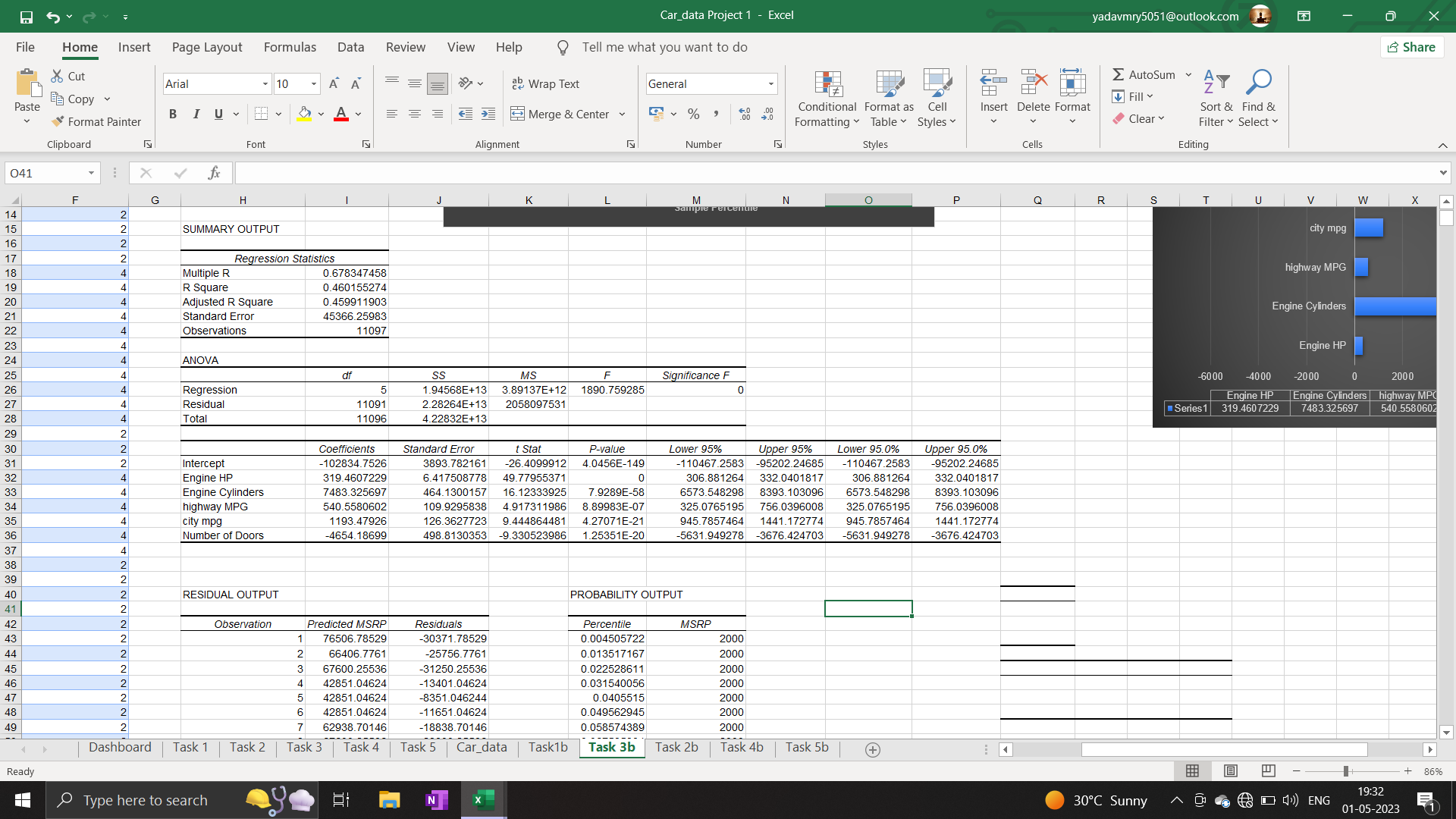
**Insight Required:** What is the relationship between a car's engine power and its price?

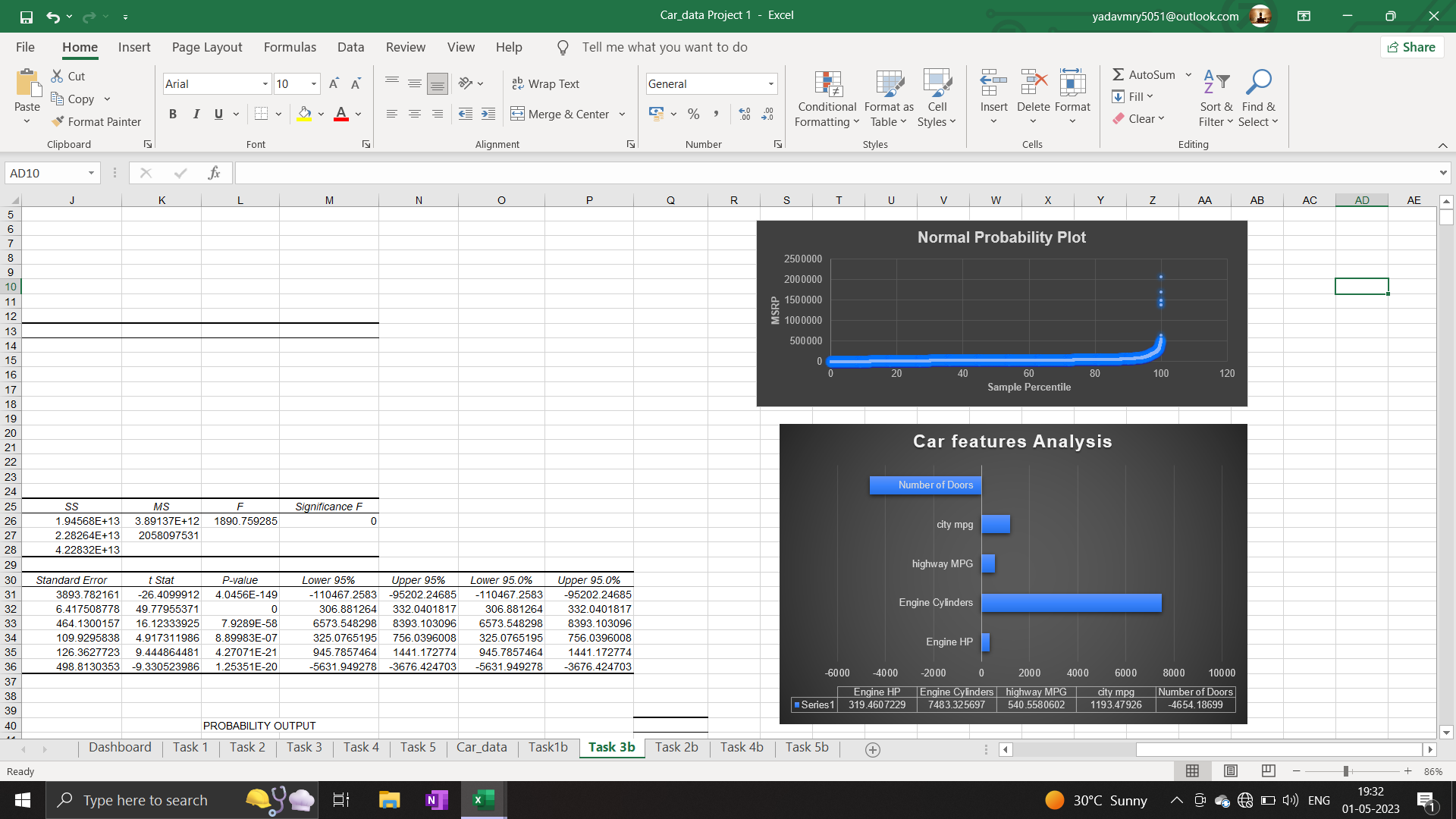
**Task 2:** Create a scatter chart that plots engine power on the x-axis and price on the y-axis. Add a trendline to the chart to visualize the relationship between these variables.



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

**Task 3:** Use regression analysis to identify the variables that have the strongest relationship with a car's price. Then create a bar chart that shows the coefficient values for each variable to visualize their relative importance.

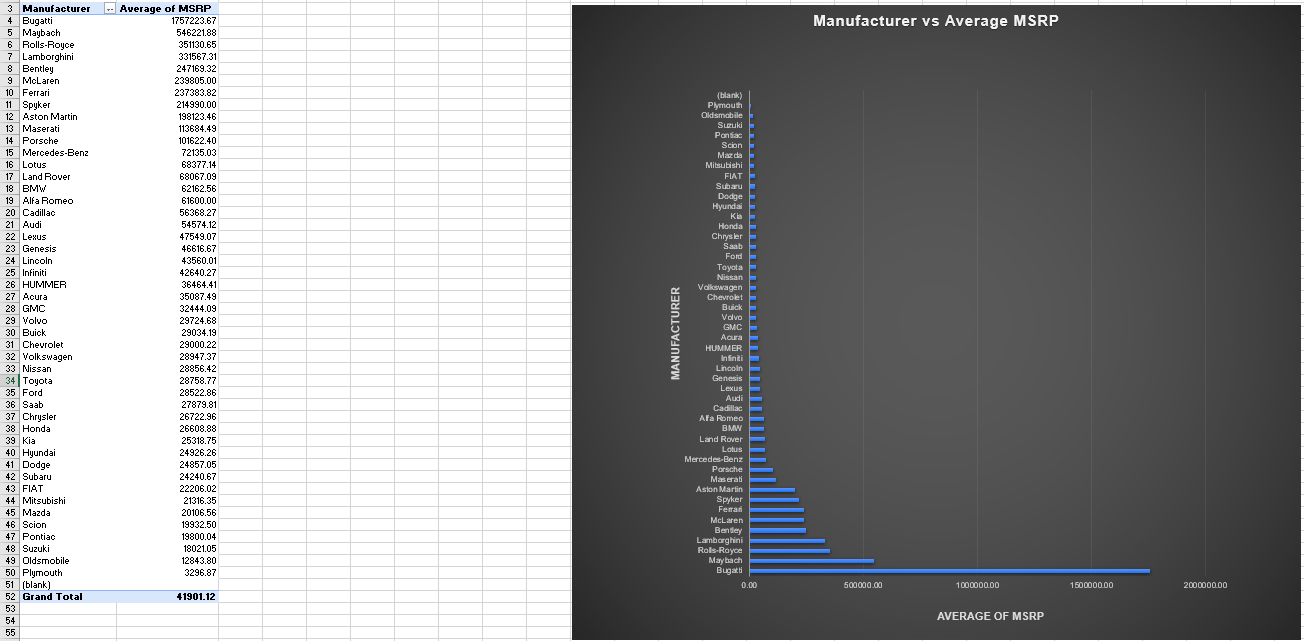




**Insight Required:** How does the average price of a car vary across different manufacturers?

**Task 4.A:** Create a pivot table that shows the average price of cars for each manufacturer.

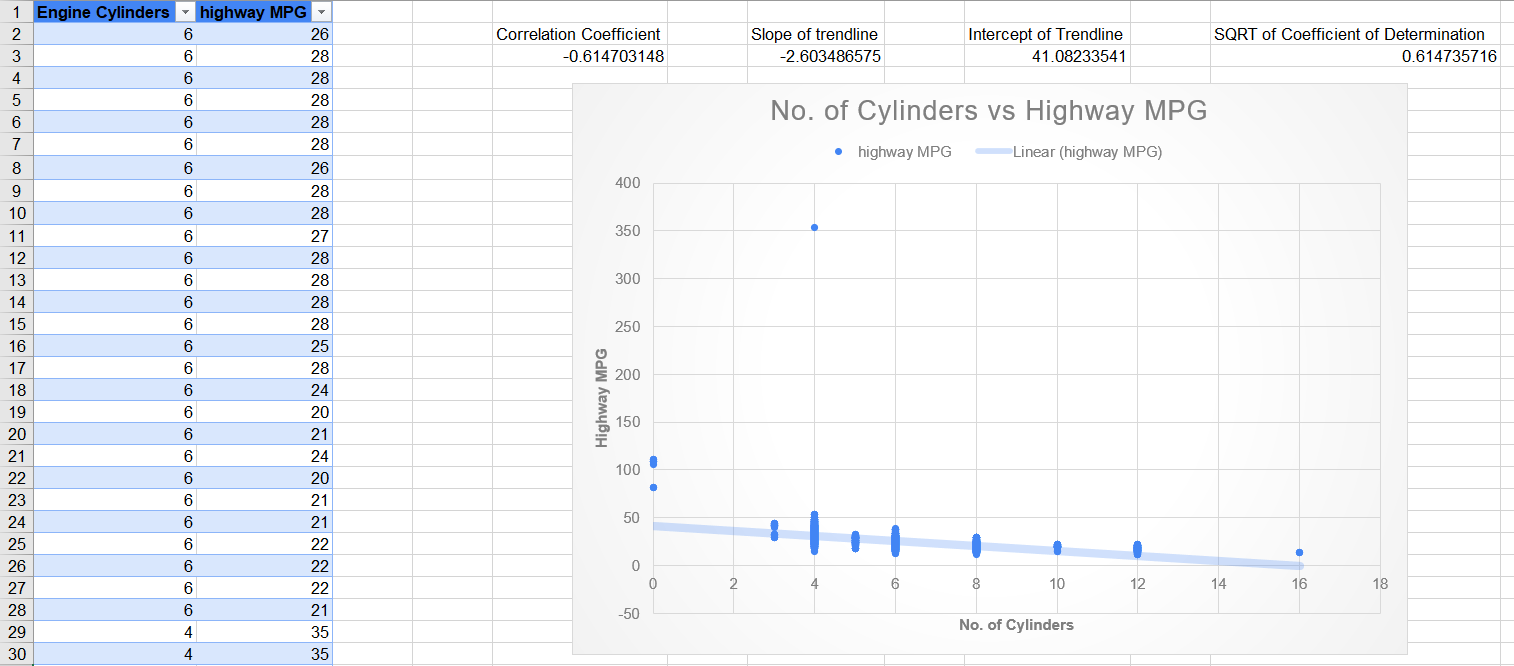
**Task 4.B:** Create a bar chart or a horizontal stacked bar chart that visualizes the relationship between manufacturer and average price.

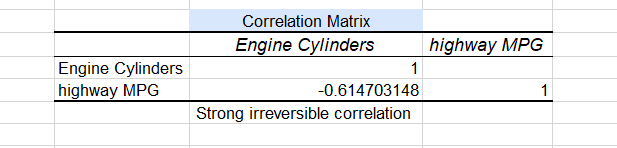


**Insight Required:** What is the relationship between fuel efficiency and the number of cylinders in a car's engine?

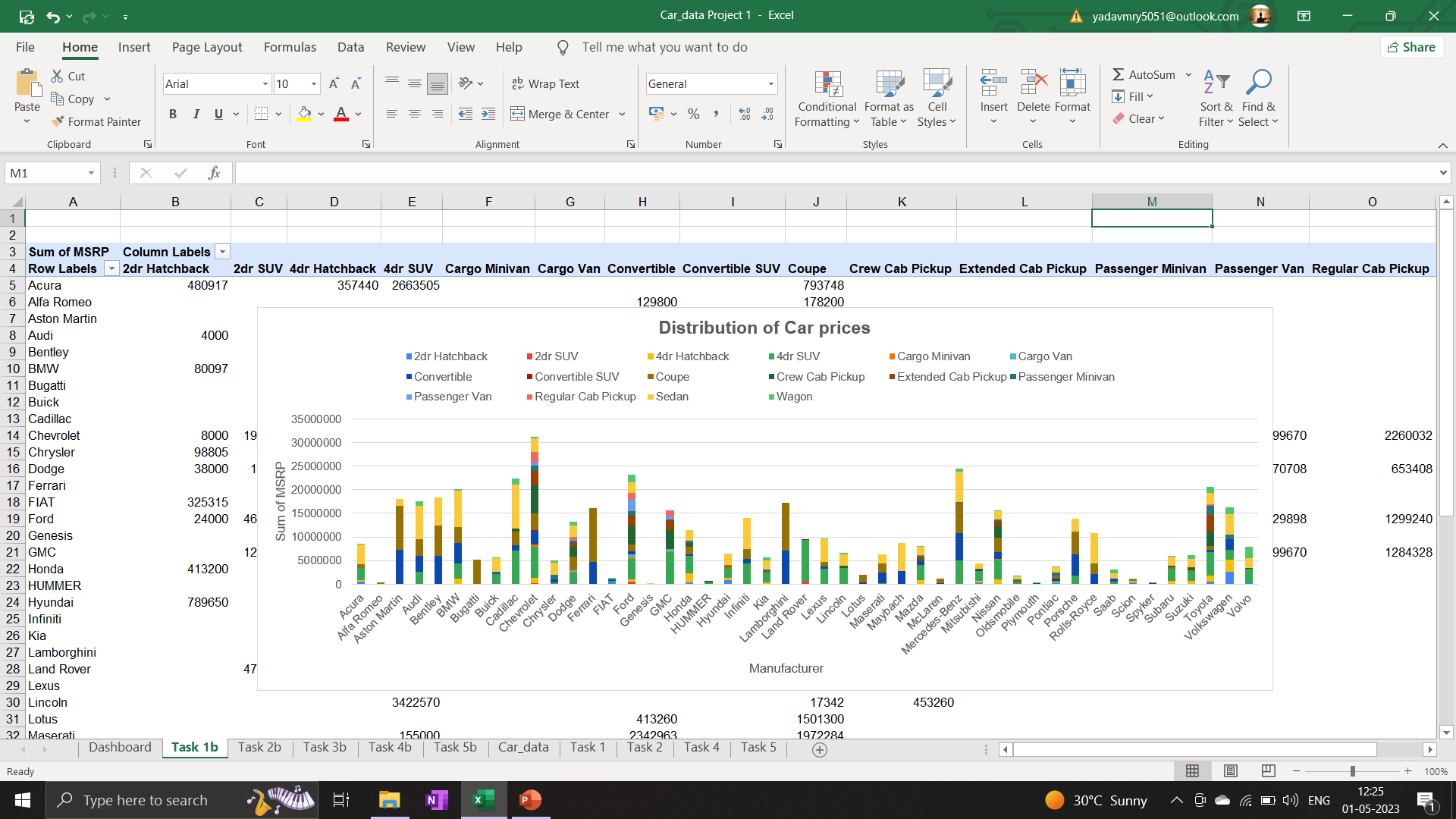
**Task 5.A:** Create a scatter plot with the number of cylinders on the x-axis and highway MPG on the y-axis. Then create a trendline on the scatter plot to visually estimate the slope of the relationship and assess its significance.

**Task 5.B:** Calculate the correlation coefficient between the number of cylinders and highway MPG to quantify the strength and direction of the relationship.

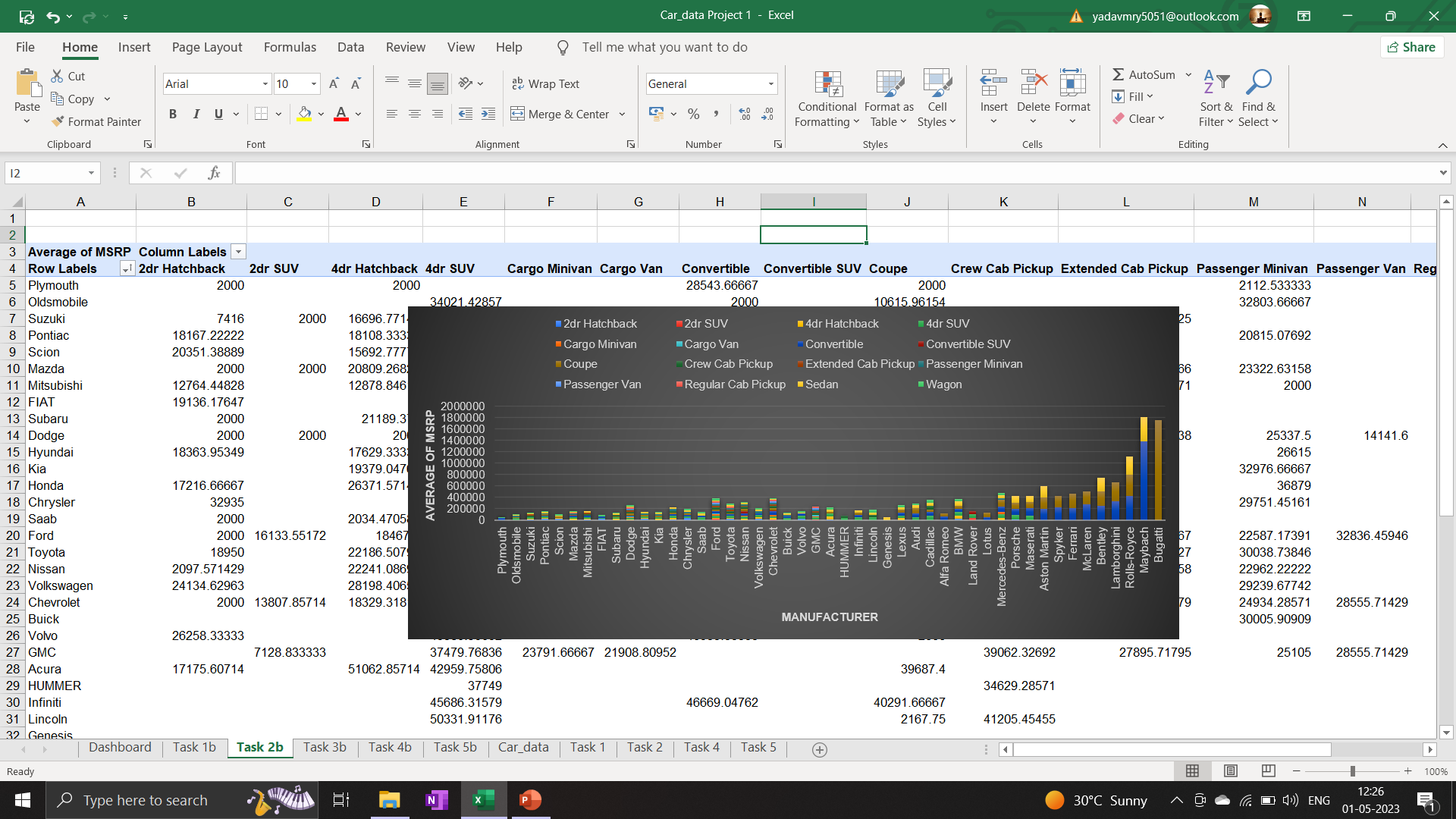




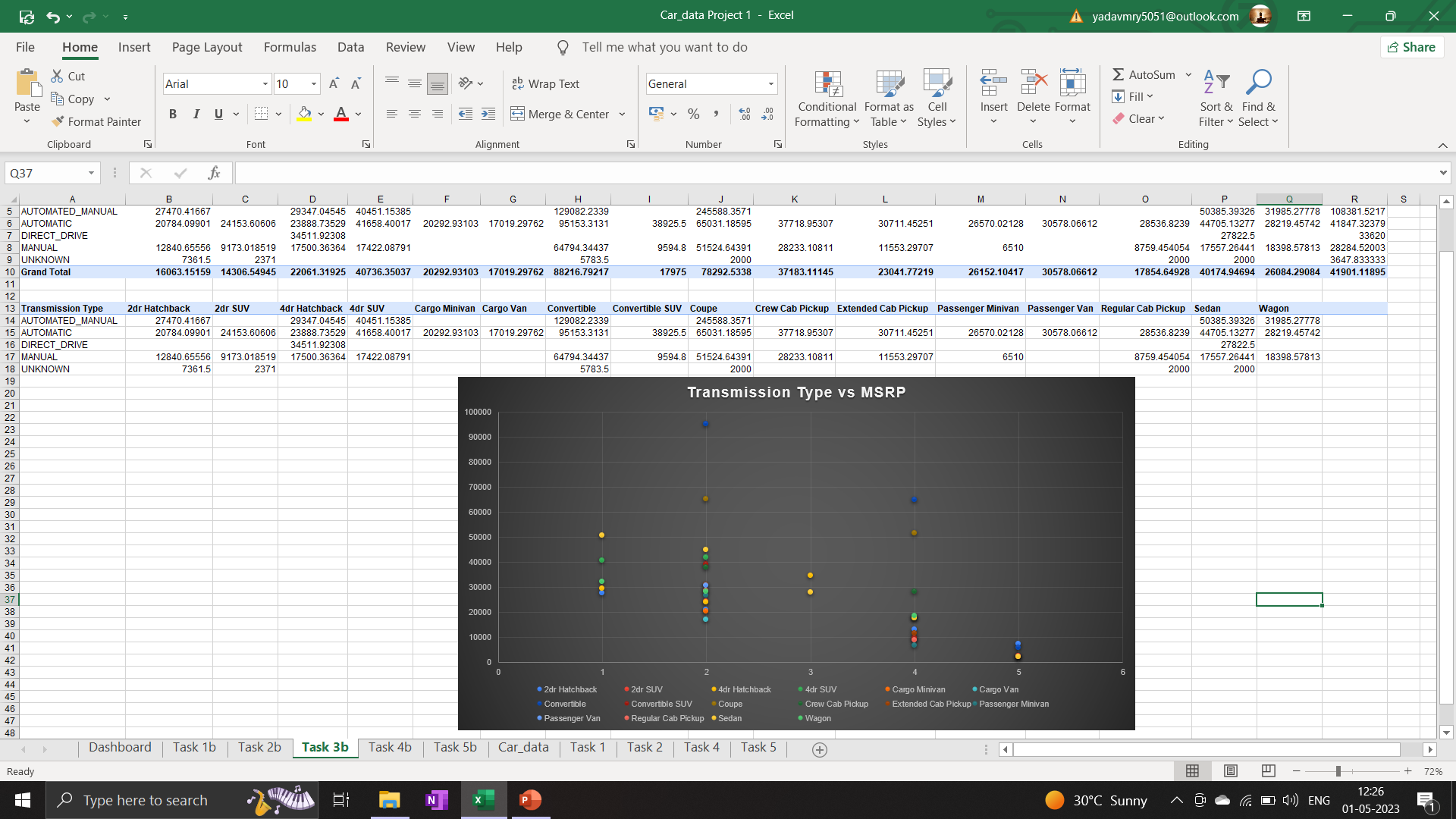
**Task 1:** How does the distribution of car prices vary by brand and body style?

d

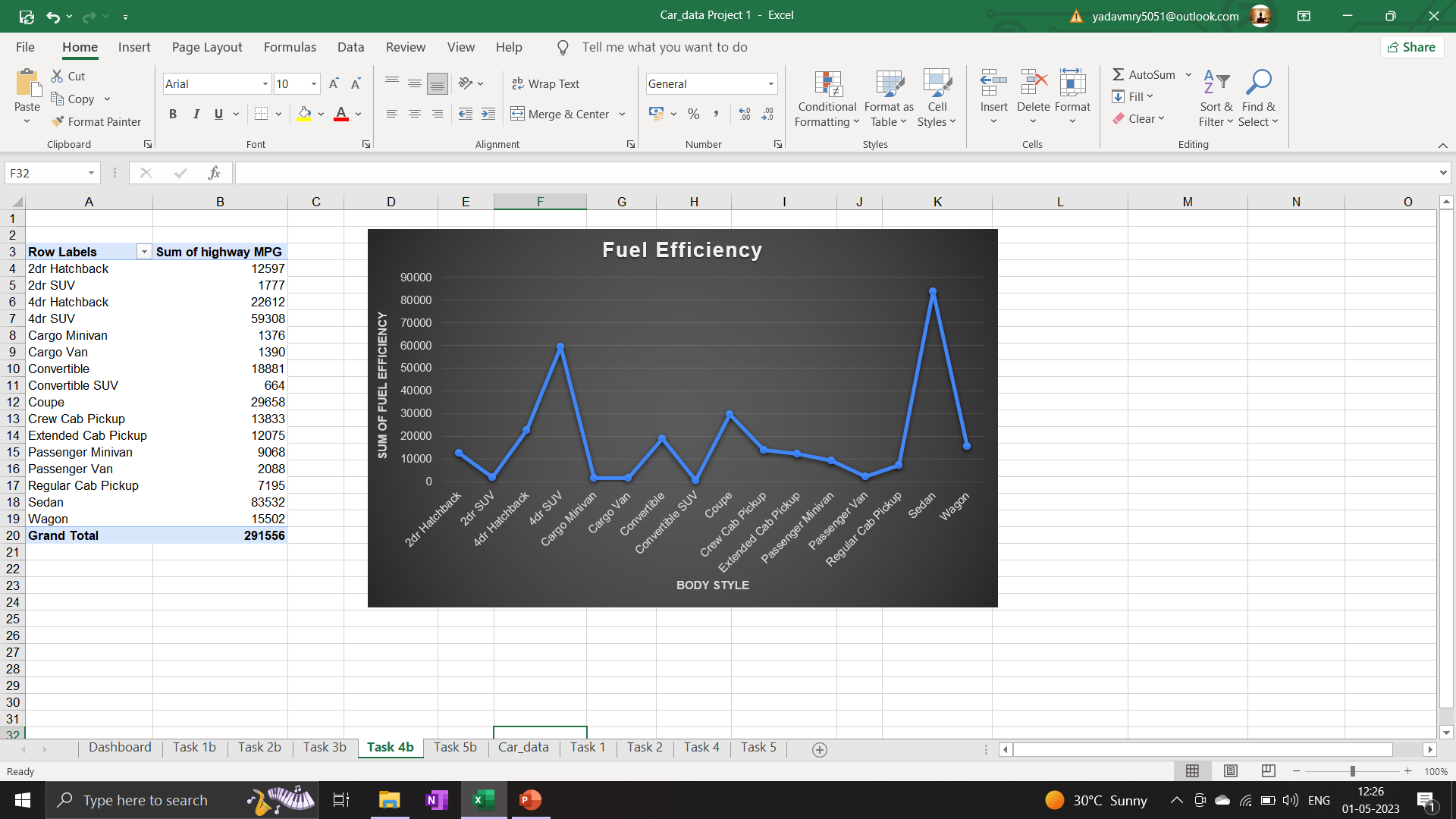
**Task 2:** Which car brands have the highest and lowest average MSRPs, and how does this vary by body style?



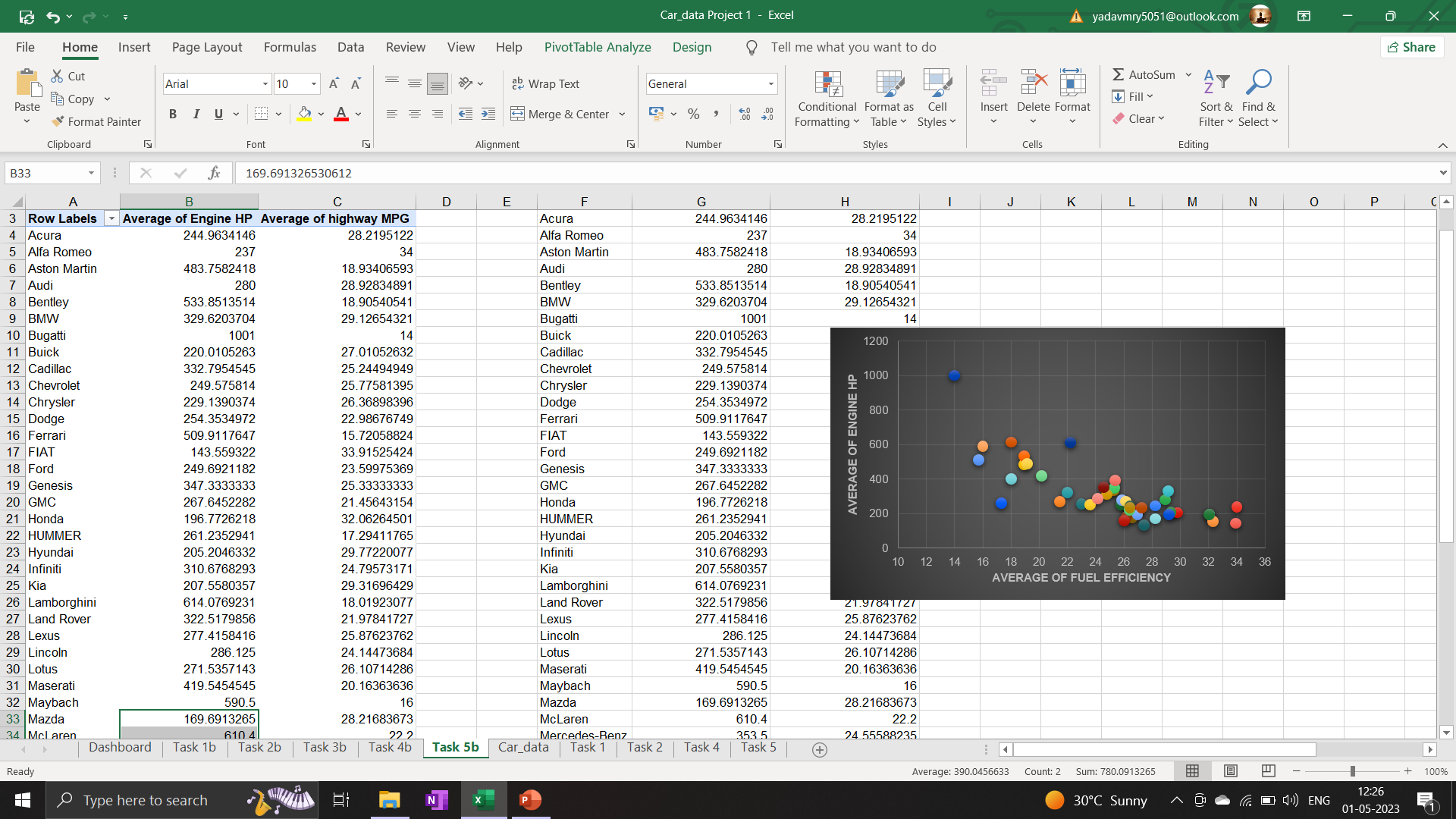
**Task 3:** How do the different feature such as transmission type affect the MSRP, and how does this vary by body style?



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



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



## 

## Dashboard:

## 

## Results:

Through this project, I learned how to use a powerful tool like Excel on a dataset to get useful insights. The many Excel functions make it easier for us to manipulate data from the data set and get various insights which can also be used for data visualization.

The hyperlink to the [Analysis\_File](https://docs.google.com/spreadsheets/d/1_Qpx_neMJbi7iC5wNjU_SOa9Ii7gGvzq/edit?usp=share_link&ouid=100333772583895800652&rtpof=true&sd=true) . By using different charts for different parameters and then building an interactive dashboard which presented all the insights from the data.

# Module 8: ABC Call Volume Trend Analysis

## Project Description:

A customer experience (CX) team consists of professionals who analyse customer feedback and data, and share insights with the rest of the organization. Typically, these teams fulfil various roles and responsibilities such as: Customer experience programs (CX programs), Digital customer experience, Design and processes, Internal communications, Voice of the customer (VoC), User experiences, Customer experience management, Journey mapping, Nurturing customer interactions, Customer success, Customer support, Handling customer data, Learning about the customer journey.

Advertising is a way of marketing your business in order to increase sales or make your audience aware of your products or services. Until a customer deal with you directly and actually buys your products or services, your advertising may help to form their first impressions of your business. Target audience for businesses could be local, regional, national or international or a mixture. So, they use different ways for advertisement. Some of the types of advertisement are: Internet/online directories, Trade and technical press, Radio, Cinema, Outdoor advertising, National papers, magazines and TV. Advertising business is very competitive as a lot of players bid a lot of money in a single segment of business to target the same audience. Here comes the analytical skills of the company to target those audiences from those types of media platforms where they convert them to their customers at a low cost.

## Approach:

Downloading the data set and performing EDA to understand the data set. Checked for the null values and the distribution. Defining the problem and to analyse each task first noted the features to be used. Now to get result I checked all the functions which will be required to perform the operations. Create charts and graphs to define underlying aspects of the data. Finally created a report consisting the description, approach, result, insights, conclusion, etc.

## Tech-Stack Used:

For this assignment I have used Microsoft Excel (2019).

## Insights & results:

**Average call time**

##### Calculate the average call time duration for all incoming calls received by agents (in each Time\_Bucket).

|  |  |
| --- | --- |
| **Row Labels** | **Average of Call\_Seconds**  **(s)** |
| 10\_11 | 203.3 |
| 11\_12 | 199.3 |
| 12\_13 | 192.9 |
| 13\_14 | 194.7 |
| 14\_15 | 193.7 |
| 15\_16 | 198.9 |
| 16\_17 | 200.9 |
| 17\_18 | 200.2 |
| 18\_19 | 202.6 |
| 19\_20 | 203.4 |
| 20\_21 | 202.8 |
| 9\_10 | 199.1 |
| **Grand Total** | **198.62** |

* + Minimum average call time is for time bucket 12\_13.



Time\_bucket vs Average call time

**203.3**

**203.4 202.8**

**202.6**

**199.3**

**200.9 200.2**

**198.9**

**199.1**

**194.7**

Total

**192.9**

**193.7**

10\_11 11\_12 12\_13 13\_14 14\_15 15\_16 16\_17 17\_18 18\_19 19\_20 20\_21 9\_10

**Time\_Bucket**

* + Total average time to answer each call is 198.62 sec.

Total numb of calls

16000

14000

14626

13313

12652

12000

11561

10561

10000

8000

6000

9588

9159

8788

8534

7238

6463

5505

4000

2000

0

9\_10 10\_11 11\_12 12\_13 13\_14 14\_15 15\_16 16\_17 17\_18 18\_19 19\_20 20\_21

Time\_Bucket

Count of Customer\_Phone\_No

numb of calls

##### 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, ….)

|  |  |  |
| --- | --- | --- |
| **Row Labels** | **Count of Customer\_Phone\_No** | **Count of Time** |
| 10\_11 | 13313 | 11.28% |
| 11\_12 | 14626 | 12.40% |
| 12\_13 | 12652 | 10.72% |
| 13\_14 | 11561 | 9.80% |
| 14\_15 | 10561 | 8.95% |
| 15\_16 | 9159 | 7.76% |
| 16\_17 | 8788 | 7.45% |
| 17\_18 | 8534 | 7.23% |
| 18\_19 | 7238 | 6.13% |
| 19\_20 | 6463 | 5.48% |
| 20\_21 | 5505 | 4.67% |
| 9\_10 | 9588 | 8.13% |
| **Grand Total** | **117988** | **100.00%** |

Numb of calls

14.0%

12.0%

12.4%

11.3%

10.7%

10.0%

9.8%

9.0%

8.1%

8.0%

7.8%

7.4%

7.2%

6.1%

6.0%

5.5%

4.7%

4.0%

2.0%

0.0%

9\_10 10\_11 11\_12 12\_13 13\_14 14\_15 15\_16 16\_17 17\_18 18\_19 19\_20 20\_21

Time\_Bucket

Count of Time

Numb of calls in the time period

##### 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.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | abandon | answered | transfer | Grand  Total |
| 01-Jan | 684 | 3883 | 77 | 4644 |
| 02-Jan | 356 | 2935 | 60 | 3351 |
| 03-Jan | 599 | 4079 | 111 | 4789 |
| 04-Jan | 595 | 4404 | 114 | 5113 |
| 05-Jan | 536 | 4140 | 114 | 4790 |
| 06-Jan | 991 | 3875 | 85 | 4951 |
| 07-Jan | 1319 | 3587 | 42 | 4948 |
| 08-Jan | 1103 | 3519 | 50 | 4672 |
| 09-Jan | 962 | 2628 | 62 | 3652 |
| 10-Jan | 1212 | 3699 | 72 | 4983 |
| 11-Jan | 856 | 3695 | 86 | 4637 |
| 12-Jan | 1299 | 3297 | 47 | 4643 |
| 13-Jan | 738 | 3326 | 59 | 4123 |
| 14-Jan | 291 | 2832 | 32 | 3155 |
| 15-Jan | 304 | 2730 | 24 | 3058 |
| 16-Jan | 1191 | 3910 | 41 | 5142 |
| 17-Jan | 16636 | 5706 | 5 | 22347 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 18-Jan | 1738 | 4024 | 12 | 5774 |
| 19-Jan | 974 | 3717 | 12 | 4703 |
| 20-Jan | 833 | 3485 | 4 | 4322 |
| 21-Jan | 566 | 3104 | 5 | 3675 |
| 22-Jan | 239 | 3045 | 7 | 3291 |
| 23-Jan | 381 | 2832 | 12 | 3225 |
| Grand Total | 34403 | 82452 | 1133 | 117988 |
| Percentage total | 29% | 70% | 1% | 100% |

Numb of calls on daily basis:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | abandon | answered | transfer | Grand  Total |
| numb of calls on daily basis | 1495.8 | 3584.9 | 49.3 | 5129.9 |

From the Question A we know that the average call duration is 198.62 sec. So total time to answer 90% of the calls will be:

𝑡𝑜𝑡𝑎𝑙 𝑡𝑖𝑚𝑒 = 𝑡𝑜𝑡𝑎𝑙 𝑛𝑢𝑚𝑏 𝑜𝑓 𝑐𝑎𝑙𝑙𝑠 ∗ 𝑎𝑣𝑒𝑟𝑎𝑔𝑒 𝑐𝑎𝑙𝑙 𝑑𝑢𝑟𝑎𝑡𝑖𝑜𝑛

|  |  |
| --- | --- |
| time required to answer 90% calls per day | 254.7001826 |

As one person works for 4.5 hours the total numb of persons required to answer 90% of calls will be:

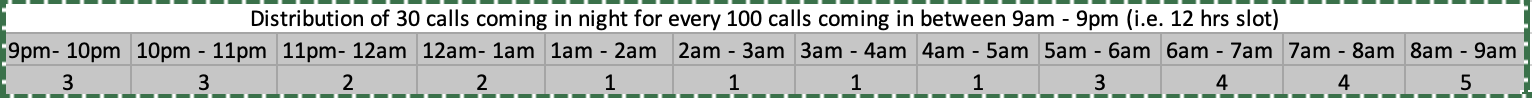
𝑛𝑢𝑚𝑏 𝑜𝑓 𝑝𝑒𝑟𝑠𝑜𝑛𝑠 =

𝑡𝑚𝑒 𝑟𝑒𝑞𝑢𝑖𝑟𝑒𝑑 𝑡𝑜 𝑎𝑛𝑠𝑤𝑒𝑟 90% 𝑜𝑓 𝑡ℎ𝑒 𝑐𝑎𝑙𝑙𝑠 ℎ𝑜𝑢𝑟𝑠 𝑓𝑜𝑟 𝑎 𝑝𝑒𝑟𝑠𝑜𝑛 𝑤𝑜𝑟𝑘𝑠

|  |  |
| --- | --- |
| numb of working persons | 57 |

##### So, to answer 90% of the calls we will require 57 agents.

##### 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%.

**Assumption:** An agent work for 6 days a week; On an average total unplanned leaves per agent is 4 days a month; An agent total working hrs is 9 Hrs out of which 1.5 Hrs goes into lunch and snacks in the office. On average an agent occupied for 60% of his total actual working Hrs (i.e., 60% of 7.5 Hrs) on call with customers/ users. Total days in a month is 30 days.

|  |  |  |
| --- | --- | --- |
|  | Values | Formulas |
| Daily call volume from (9AM-9PM) | 5129.9 | Average of the total numb of calls per day |
| Total calls from (9PM-9AM) | 1539.0 | Average of the total numb of calls per night |
| Total night time required | 76.4 | night time to answer 90% of calls(hours) =  numb of calls at night ∗ average call duration ∗ 0.9  =  3600 |
| Additional man power needed | 17 | total night time  Additional man power =  hours for a agent works |
| Total man power needed | 74 | Total man power = agents on day time + agents on night  time |

##### Total agents needed to answer night calls is 17.

## Challenges:

The project consists of the deep knowledge of the call volume trend and analysis of the company. Also, it includes the prediction of number of agents at day and night time which was a challenging part.

# Appendix

#### The Google drive links of the following projects is given below:

#### Data Analytics Process - [Link](https://drive.google.com/file/d/1otmdQ_EbkBljN8EhqNbn-eg0uGdoNTzq/view?usp=share_link)

#### Instagram User Analytics - [Link](https://drive.google.com/file/d/1ISB796obpJ5XEYf7G8m3rH4Bfojoy6fC/view?usp=share_link)

#### Operation & Metric Analytics - [Link](https://drive.google.com/file/d/1e43U2TpiBgCQOH3QWtN7Z_iKsU0sbKhq/view?usp=share_link)

#### Hiring Process Analytics - [Link](https://drive.google.com/file/d/1o-3dmWuMYAk0l0hl-pilPVTMCxd07vCN/view?usp=share_link)

#### IMDB Movie Analysis - [Link](https://drive.google.com/file/d/11YYatS6sYe4EJf6mNh60CQIVMMGw6bn6/view?usp=share_link)

#### Bank Loan Case Study - [Link](https://drive.google.com/file/d/1F1Cy2L9kJAl4J7WgI1s181Jq0Lp8ZL1_/view?usp=share_link)

#### Impact of Car Features - [Link](https://drive.google.com/drive/folders/10ChneN1v1t3k7GuHaK6z3jJRq77B1PmB?usp=share_link)

#### ABC Call Volume Trend - [Link](https://drive.google.com/file/d/10KH8Dx4skccZb6Xu1AafUmNis0z4abEH/view?usp=share_link)