PORTFOLIO

SNEHA KATOLE

Professional Background

I am Sneha Katole, currently pursuing a Bachelor of Technology at Vishwakarma Institute of Technology, Pune, with an expected graduation in 2027. I have maintained a strong academic record with a current CGPA of 8.71.

My professional interests lie in the fields of:

- Data Analysis & Visualization
- Data Science & Machine Learning
- Relational Databases (MySQL)
- Excel-based Statistical Modeling

I am deeply passionate about turning raw data into meaningful insights that drive smarter decisions. I have developed hands-on projects that demonstrate the application of analytical thinking, from identifying customer behavior patterns to optimizing operations through data-driven planning.

In addition to my technical skills, I bring strong adaptability, curiosity, and a collaborative spirit. I enjoy exploring new tools, frameworks, and approaches to solve challenging problems. I am particularly interested in domains like customer experience analytics, business intelligence, and operations research.

I aspire to grow as a data professional who not only understands the tools but also the why behind the data — using analytics to bridge the gap between information and impact.

Table of Contents

What is Data Analytics

Data analytics is the process of examining, organizing, and interpreting raw data to uncover meaningful patterns, trends, and insights. It involves the use of statistical techniques, programming tools, and visualization methods to support decision-making, optimize operations, and solve business problems. By transforming complex datasets into actionable information, data analytics empowers organizations to make informed choices, predict future outcomes, and gain a competitive advantage in today's data-driven world.

As businesses continue to generate vast amounts of data through digital interactions, the demand for skilled data analysts has grown rapidly. From improving customer experience to enhancing operational efficiency, data analytics plays a vital role across industries such as finance, healthcare, marketing, and technology. Whether through descriptive dashboards or predictive models, data analysts bridge the gap between raw information and strategic decision-making, turning data into a powerful asset for innovation and growth.

Data Analytics Process

Module- I

Description:

We use Data Analytics in everyday life without even knowing it.

For eg : Going to a market to buy something .

1. **Plan: We first decide which things I need before going to market . Is it a shirt , jeans , footwear etc.**
2. **Prepare: Next I need to check how much I am willing to spend and how to get that money.**
3. **Process: Then I need to check how much I want from the data. Like if I am going to buy footwear what do I want - slippers / shoes / sandals etc.**
4. **Analyze: You obviously won't buy things which are out of trend, Also you need to check does the jeans which you have and the color of t-shirt you want to buy, will it make a good combination.**
5. **Share: Now you communicate your idea to the shopkeeper to find the best suitable fit for you.**
6. **Act: Then you finally buy it!**

Task:
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. You can prepare a PPT/PDF on a real-life scenario explaining it with the above process (Plan, Prepare, Process, Analyze, Share, Act) and submit it as part of this task.

Case Study: Understanding the aΣendance trends for a school

1. Plan: first understanding which parameters are affecϴng the attendance records. For example, the grades of the student, health condition of a student, weather, etc.

2. Prepare: next get the data regarding the attendance records from the school for a sample of students.

3. Process: cleaning the missing records, removing the errors (where student present has been marked absent or vice versa cause of manual errors during attendance filling by the teacher).

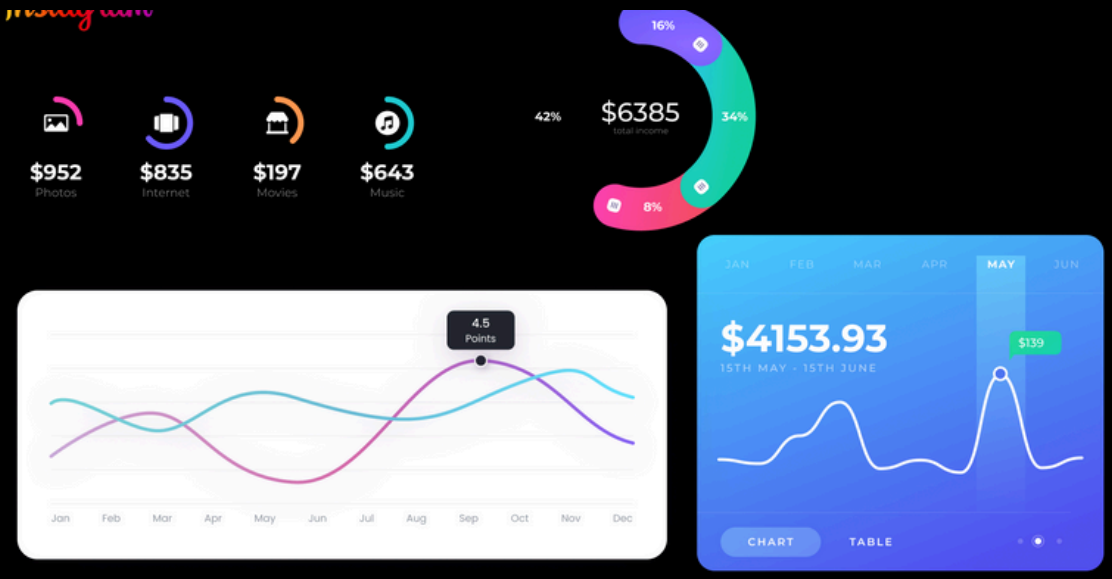
4. Analyse: understand the co-relation of different factors discussed in the planning stage with the attendance trends and how they're affecting the attendance of students.

5. Share: Communicate the insights gained to the school, with the parents of the student with low attendance record and discussing the reason accordingly.

6. Act: Then take actions regarding what could be done to reduce the low attendance.

Instagram User Analytics

Module- II



Project Description

This project focuses on analysing Instagram user interactions and engagement using MySQL Workbench. The goal is to extract meaningful insights that support the product and marketing teams in making informed decisions about user retention, campaign strategies, and platform improvements.

Approach

1. Explored the database structure, including tables like users, photos, likes, tags, photo_tags, comments, and follows.
2. Created SQL queries to answer the given tasks.
3. Analysed outputs to derive actionable insights as asked.
4. Compiled queries, outputs, and conclusions in a structured report.

Tech-Stack Used

1. MySQL Workbench: Used for writing and executing SQL queries.
2. SQL: Structured Query Language for data extraction and analysis.

Queries and Output

```
SELECT * FROM users;
select * from photos;
select * from comments;
select * from likes;
select * from follows;
select * from tags;
select * from photo_tags;

/*marketing analysis
loyal user reward(five oldest users)
inactive user engagement(users with no photo post)
contest winner declaration(details of user with most likes on the single photo)
hashtag research(top five used hastags)
ad campaign launch(day of the week with most user reg)
*/
/*investor metrics
user engagement(avg posts/user and total photos/total users)
bots and fake accounts(users who liked every single photo)
*/
```

Marketing Analysis

```
-- loyal user reward
select username, created_at
from users
order by created_at asc
limit 5;

-- inactive user engagement
select u.username
from users u
left join photos p on u.id=p.user_id
where p.id is null;

-- contest winner declaration
select u.username, p.id, count(l.photo_id) as like_count
from photos p
join likes l on p.id=l.photo_id
join users u on p.user_id=u.id
group by p.id,u.username
order by like_count desc
limit 1;
```

```sql
-- hashtag research
select h.tag_name, count(*) as tag_count
from photo_tags pt
join tags h on pt.tag_id=h.id
group by h.tag_name
order by tag_count desc
limit 5;

-- ad campaign launch
select dayname(created_at) as registration_day, count(*) as user_count
from users
group by registration_day
order by user_count desc
limit 1;
```

```sql
-- loyal user reward
select username, created_at
from users
order by created_at asc
limit 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fet

username	created_at
Darby_Herzog	2016-05-06 00:14:21
Emilio_Bernier52	2016-05-06 13:04:30
Elenor88	2016-05-08 01:30:41
Nicole71	2016-05-09 17:30:22
Jordyn.Jacobson2	2016-05-14 07:56:26

```sql
-- inactive user engagement
select u.username
from users u
```

Result Grid | Filter Rows: | Export: | Wrap Cell Con

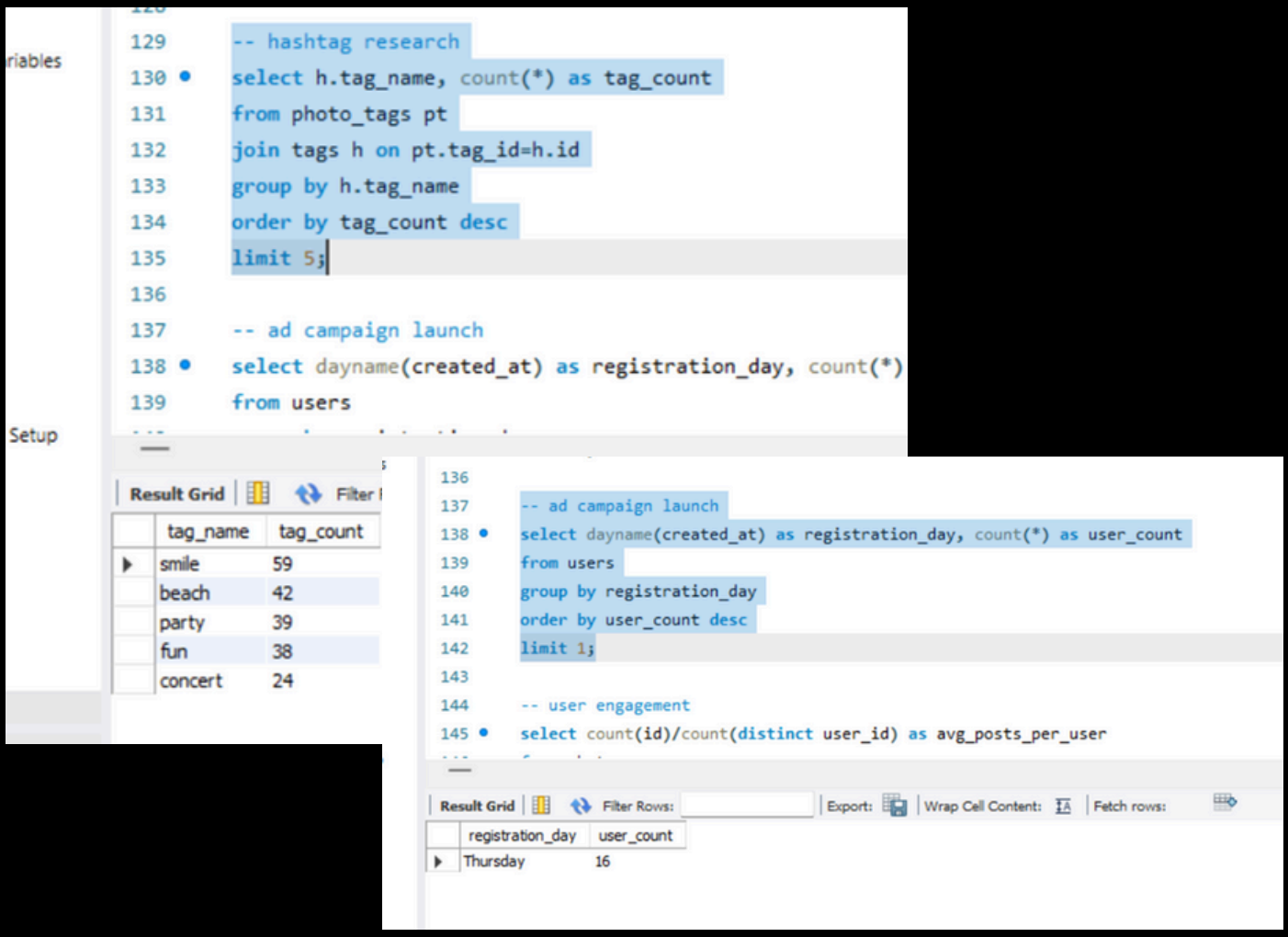
username
Jaclyn81
Rocio33
Maxwell.Halvorson
Tierra.Trantow
Pearl7
Ollie_Ledner37
Mckenna17
David.Osinski47
Morgan.Kassulke
Linnea59
Duane60
Julien_Schmidt
Mike.Auer39
Franco_Keebler64
Nia_Haag
Hulda.Macejkovic
Leslie67
Janelle.Nikolaus81
Darby_Herzog
Esther.Zulauf61
Bartholome.Bernhard
Jessyca_West
Esmeralda.Mraz57
Bethany20

```sql
-- contest winner declaration
select u.username, p.id, count(l.photo_id) as like_co
from photos p
join likes l on p.id=l.photo_id
join users u on p.user_id=u.id
group by p.id,u.username
order by like_count desc
limit 1;

-- hashtag research
select h.tag_name, count(*) as tag_count
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

username	id	like_count
Zack_Kemmer93	145	48



Investor Metrics

```sql
-- user engagement
select count(id)/count(distinct user_id) as avg_posts_per_user
from photos;

select (select count(*) from photos)/(select count(*) from users)
as photo_per_user;

-- bots and fake accounts
select u.username, count(l.photo_id) as total_likes
from users u
join likes l on u.id=l.user_id
group by u.id
having total_likes= (select count(*) from photos);
```

```sql
-- user engagement
select count(id)/count(distinct user_id) as avg_posts_per_user
from photos;

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg_posts_per_user
3.4730

```sql
from photos;

select (select count(*) from photos)/(select count(*) from users)
as photo_per_user;

-- bots and fake accounts
select u.username, count(l.photo_id) as total_likes
from users u
join likes l on u.id=l.user_id
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

photo_per_user
2.5700

```sql
-- bots and fake accounts
select u.username, count(l.photo_id) as total_likes
from users u
join likes l on u.id=l.user_id
group by u.id
having total_likes= (select count(*) from photos);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

username	total_likes
Aniya_Hackett	257
Jaclyn81	257
Rocio33	257
Maxwell.Halvorson	257
Ollie_Ledner37	257
Mckenna17	257
Duane60	257
Julien_Schmidt	257
Mike.Auer39	257
Nia_Haag	257
Leslie67	257
Janelle.Nikolaus81	257
Bethany20	257

Insights

1. **Most loyal users can be approached for rewards or brand ambassador opportunities.**
2. **For inactive users, targeted email campaigns should be made to encourage them to share their first post.**
3. **Promoting the content of contest winner for user with most likes on a single photo can inspire others to participate in similar contest.**
4. **Hashtag research helps partner brands align their marketing strategies.**
5. **Ad campaign launch analytics could help team schedule ads and running those in high-traffic days.**
6. **User engagement helps in taking actions regarding feature improvements and content incentives.**
7. **Bot accounts detection helps in tacking fake accounts, helping authentic engagement and platform integrity.**

Results

1. **Successfully identified key user behaviours.**
2. **Provided actionable insights for marketing and product teams.**
3. **Highlighted areas of concern for investor reporting.**

Operation Analytics with Metric Spike

Module -III

Case Study 1: Job Data Analysis

Project Description:
The project stresses on analysis of job data for insights into the job review patterns, throughput, and language share analysis.

Approach:

1. Created the table job_data with columns job_id, actor_id, event, language, time_spent, org, and ds using the database case_study1.
2. Entries are added randomly utilizing the SOL functions.
3. Executed SQL queries for each task.
4. Reviewed obtained outputs to identify patterns.
5. Drew insights and interpretations from the results.

Tech-Stack Used:

1. MySQL Workbench: For executing the SQL queries.
2. SQL: For data analysis.

Queries and Output

```sql
-- job data analysis
show databases;
create database case_study1;
use case_study1;

CREATE TABLE job_data (
    job_id INT PRIMARY KEY,
    actor_id INT,
    event VARCHAR(20),
    language VARCHAR(20),
    time_spent INT,
    org VARCHAR(50),
    ds DATE
);
```

```sql
-- Insert 10,000 random rows into job_data
INSERT INTO job_data (job_id, actor_id, event, language, time_spent, org, ds)
SELECT
    t.n,
    FLOOR(1 + (RAND() * 1000)),
    ELT(FLOOR(1 + (RAND() * 3)), 'decision', 'skip', 'transfer'),
    ELT(FLOOR(1 + (RAND() * 5)), 'English', 'Spanish', 'French', 'German', 'Chinese'),
    FLOOR(1 + (RAND() * 300)),
    ELT(FLOOR(1 + (RAND() * 5)), 'OrgA', 'OrgB', 'OrgC', 'OrgD', 'OrgE'),
    DATE_ADD('2020-11-01', INTERVAL FLOOR(RAND() * 30) DAY)
FROM (
    SELECT a.N + b.N * 10 + c.N * 100 + d.N * 1000 AS n
    FROM
        (SELECT 0 N UNION SELECT 1 UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8 UNI
        (SELECT 0 N UNION SELECT 1 UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8 UNI
        (SELECT 0 N UNION SELECT 1 UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8 UNI
        (SELECT 0 N UNION SELECT 1 UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8 UNI
    LIMIT 100000
) t;
```

```sql
select * from job_data limit 0,10000;

-- jobs reviewed over time
-- throughput analysis
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Conten

job_id	actor_id	event	language	time_spent	org	ds
9983	888	transfer	Spanish	89	OrgD	2020-11-24
9984	197	decision	English	252	OrgA	2020-11-25
9985	285	transfer	Spanish	221	OrgE	2020-11-19
9986	669	decision	English	245	OrgD	2020-11-17
9987	832	decision	German	230	OrgD	2020-11-12
9988	201	decision	Chinese	286	OrgB	2020-11-10
9989	340	decision	German	234	OrgE	2020-11-20
9990	488	decision	German	245	OrgD	2020-11-30
9991	625	decision	French	92	OrgB	2020-11-14
9992	478	skip	English	164	OrgD	2020-11-29
9993	491	decision	Chinese	52	OrgB	2020-11-19
9994	68	transfer	French	279	OrgB	2020-11-04
9995	302	skip	Spanish	61	OrgA	2020-11-27
9996	587	decision	French	188	OrgD	2020-11-21
9997	960	skip	English	81	OrgA	2020-11-17
9998	200	skip	Spanish	39	OrgE	2020-11-05
9999	781	decision	Spanish	54	OrgE	2020-11-07
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- throughput analysis
-- language share analysis
-- duplicate rows detection

-- jobs reviewed/day for nov 2020
SELECT
    ds,
    COUNT(job_id) AS jobs_reviewed
FROM job_data
WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds
ORDER BY ds;

-- 7-day rolling avg of throughput or daily metric
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ds	jobs_reviewed
2020-11-01	331
2020-11-02	347
2020-11-03	358
2020-11-04	339
2020-11-05	343
2020-11-06	332
2020-11-07	354
2020-11-08	336
2020-11-09	304

```sql
-- 7-day rolling avg of throughput or daily metric
WITH daily_events AS (
    SELECT
        ds,
        COUNT(*) / 86400 AS events_per_second
    FROM job_data
    GROUP BY ds
)
SELECT
    ds,
    AVG(events_per_second) OVER (
        ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW
    ) AS rolling_avg_throughput
FROM daily_events;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ds	rolling_avg_throughput
2020-11-01	0.00380000
2020-11-02	0.00390000
2020-11-03	0.00396667
2020-11-04	0.00395000
2020-11-05	0.00396000
2020-11-06	0.00393333
2020-11-07	0.00395714
2020-11-08	0.00397143
2020-11-09	0.00390000
2020-11-10	0.00392857
2020-11-11	0.00391429

Result 26

```sql
-- percentage share/language over last 30 days
WITH last_30_days AS (
    SELECT language, COUNT(*) AS language_count
    FROM job_data
    WHERE ds >= DATE_SUB('2020-11-30', INTERVAL 30 DAY)
    GROUP BY language
)
SELECT
    language,
    language_count,
    (language_count * 100.0) / SUM(language_count) OVER () AS l
FROM last_30_days
ORDER BY language_share_percentage DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

language	language_count	language_share_percentage
French	2036	20.33966
Chinese	2020	20.17982
German	1993	19.91009
English	1992	19.90010
Spanish	1969	19.67033

```sql
-- display duplicate rows
SELECT
    job_id, actor_id, event, language, time_spent, org, ds,
    COUNT(*) AS duplicate_count
FROM job_data
GROUP BY job_id, actor_id, event, language, time_spent, org, ds
HAVING COUNT(*) > 1;
-- no duplicates in the dataset
-- adding the duplicates
-- Insert 10 duplicate rows into job_data
-- Insert EXACT duplicates 10 times
INSERT INTO job_data (job_id, actor_id, event, language, time_spent, org, ds)
VALUES
(11001, 201, 'decision', 'English', 120, 'OrgA', '2020-11-10'),
(11002, 202, 'skip', 'Spanish', 80, 'OrgB', '2020-11-11'),
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

job_id	actor_id	event	language	time_spent	org	ds	duplicate_count

```sql
(11002, 202, 'skip', 'Spanish', 80, 'OrgB', '2020-11-11'),
(11003, 203, 'transfer', 'French', 150, 'OrgC', '2020-11-12'),
(11004, 204, 'decision', 'German', 200, 'OrgD', '2020-11-13'),
(11005, 205, 'skip', 'Chinese', 90, 'OrgE', '2020-11-14');

-- checking for duplicates again
SELECT
    actor_id, event, language, org, ds,
    COUNT(*) AS duplicate_count
FROM job_data
GROUP BY actor_id, event, language, org, ds
HAVING duplicate_count > 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

actor_id	event	language	org	ds	duplicate_count
61	decision	Spanish	OrgE	2020-11-08	2
958	transfer	German	OrgE	2020-11-12	2
784	transfer	Spanish	OrgE	2020-11-09	2
902	decision	French	OrgC	2020-11-01	2
987	transfer	French	OrgE	2020-11-23	2
761	skip	English	OrgC	2020-11-11	2
451	decision	Chinese	OrgC	2020-11-14	2
98	transfer	Spanish	OrgC	2020-11-23	2
686	transfer	French	OrgE	2020-11-06	2
286	skip	Chinese	OrgA	2020-11-08	2
498	transfer	Spanish	OrgE	2020-11-20	2
758	decision	English	OrgC	2020-11-09	2
532	skip	Spanish	OrgC	2020-11-20	2
701	skip	Chinese	OrgA	2020-11-06	2
314	decision	French	OrgA	2020-11-18	2

Result 29

Insights:

1. **Jobs reviewed/day: the daily review count shows job activity patterns. Spikes indicate high-priority days/ system anomalies while dips highlight holidays/outages.**
2. **Throughput analysis: the rolling average overcome daily fluctuations, hence providing long-term patterns. It is preferred over daily metrics for identifying consistent patterns whereas in case of spotting sudden spikes or dips, daily metric should be preferred.**
3. **Language share analysis: highest for French, least for Spanish.**
4. **Duplicate rows detection: for data inconsistency identification. There were no duplicate rows initially, so after inserting exact duplicates, checking whether the query works correctly. It did.**

Results:

1. **Developed and executed SQL queries for all tasks.**
2. **Provided insights into job review patterns, throughput trends, and language shares.**
3. **Validated duplicate detection with sample data.**

Case Study 2: Investigating Metric Spike

Project description:
This project focuses on analysing user engagement, growth, retention, and email activity to identify and investigate metric spikes. The goal is to uncover meaningful insights that can guide business decisions and product improvements. SQL queries were used to extract and analyse data from three key datasets: users, events, and email_events.

Approach:
1.Data Exploration: Reviewed the structure of the three datasets:
- **users: Contains user information, including signup dates.**
- **events: Logs user actions, including timestamps and device information.**
- **email_events: Tracks email interactions with columns for user actions and timestamps.**

2.SQL Query Development: Constructed queries to address key metrics such as user engagement, growth, and email interaction.
3.Results Analysis: Evaluated outputs for trends, spikes, and anomalies.
4.Report Compilation: Documented queries, outputs, and insights.

Tech-stack used:
1. **MySQL Workbench: For writing and executing SQL queries.**
2. **SQL: Structured Query Language for data analysis**

Queries and Output

```sql
-- weekly user engagement
SELECT
    YEAR(e.occured_at) AS year,
    WEEK(e.occured_at) AS week,
    COUNT(DISTINCT e.user_id) AS active_users
FROM events e
GROUP BY year, week
ORDER BY year, week;
```

Result Grid | Filter Rows:

year	week	active_users
2014	17	663
2014	18	1068
2014	19	1113
2014	20	1154
2014	21	1121
2014	22	1186
2014	23	1232
2014	24	1275
2014	25	1264
2014	26	1302
2014	27	1372
2014	28	1365
2014	29	1376
2014	30	1467
2014	31	1299
2014	32	1225
2014	33	1225
2014	34	1204
2014	35	104

```sql
-- user growth analysis
SELECT
    YEAR(created_at) AS year,
    MONTH(created_at) AS month,
    COUNT(user_id) AS new_users
FROM users
GROUP BY year, month
ORDER BY year, month;
```

Result Grid | Filter Rows: | Export:

year	month	new_users
2013	1	160
2013	2	160
2013	3	150
2013	4	181
2013	5	214
2013	6	213
2013	7	284
2013	8	316
2013	9	330
2013	10	390
2013	11	399
2013	12	486
2014	1	552
2014	2	525
2014	3	615
2014	4	726
2014	5	779
2014	6	873
2014	7	997
2014	8	1031

Result 2

```sql
-- weekly retention analysis
WITH signup_cohort AS (
    SELECT
        user_id,
        DATE(created_at) AS signup_date
    FROM users
),
weekly_engagement AS (
    SELECT
        u.user_id,
        s.signup_date,
        YEAR(e.occured_at) AS year,
        WEEK(e.occured_at) AS week
    FROM events e
    JOIN users u ON e.user_id = u.user_id
    JOIN signup_cohort s ON u.user_id = s.user_id
)
SELECT
    signup_date,
    year,
    week,
    COUNT(DISTINCT us
FROM weekly_engagemer
GROUP BY signup_date,
ORDER BY signup_date,
```

signup_date	year	week	retained_users
2013-01-01	2014	17	1
2013-01-01	2014	18	1
2013-01-01	2014	19	2
2013-01-01	2014	20	2
2013-01-01	2014	21	1
2013-01-01	2014	22	1
2013-01-01	2014	23	1
2013-01-01	2014	24	2
2013-01-01	2014	25	2
2013-01-01	2014	26	1
2013-01-01	2014	27	1
2013-01-01	2014	30	2
2013-01-01	2014	31	1
2013-01-02	2014	17	1
2013-01-02	2014	18	2
2013-01-02	2014	19	1
2013-01-02	2014	20	1
2013-01-02	2014	21	1
2013-01-02	2014	22	2
2013-01-02	2014	23	2

```sql
-- weekly engagement per device
SELECT
    YEAR(e.occured_at) AS year,
    WEEK(e.occured_at) AS week,
    e.device,
    COUNT(DISTINCT e.user_id) AS active_users
FROM events e
GROUP BY year, week, e.device
ORDER BY year, week, active_users DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	week	device	active_users
2014	17	macbook pro	143
2014	17	lenovo thinkpad	86
2014	17	iphone 5	65
2014	17	macbook air	54
2014	17	samsung galaxy s4	52
2014	17	dell inspiron notebook	46
2014	17	iphone 5s	42
2014	17	nexus 5	40
2014	17	ipad air	27
2014	17	asus chromebook	21
2014	17	iphone 4s	21
2014	17	acer aspire notebook	20
2014	17	ipad mini	19
2014	17	dell inspiron desktop	18
2014	17	nexus 7	18
2014	17	nokia lumia 635	17
2014	17	htc one	16
2014	17	nexus 10	16

```sql
-- email engagement analysis
SELECT DISTINCT action
FROM email_events;

SELECT
    DATE(occured_at) AS email_date,
    COUNT(*) AS total_emails_sent,
    SUM(CASE WHEN action = 'email_open' THEN 1 ELSE 0 END) AS emails_opened,
    SUM(CASE WHEN action = 'email_clickthrough' THEN 1 ELSE 0 END) AS emails_clicked,
    ROUND((SUM(CASE WHEN action = 'email_open' THEN 1 ELSE 0 END) * 100.0) / COUNT(*), 2) AS open_rate,
    ROUND((SUM(CASE WHEN action = 'email_clickthrough' THEN 1 ELSE 0 END) * 100.0) / COUNT(*), 2) AS click_rate
FROM email_events
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

email_date	total_emails_sent	emails_opened	emails_clicked	open_rate	click_rate
2014-05-01	680	145	61	21.32	8.97
2014-05-02	704	142	82	20.17	11.65
2014-05-03	73	23	23	31.51	31.51
2014-05-04	68	22	21	32.35	30.88
2014-05-05	1164	255	115	21.91	9.88
2014-05-06	757	168	82	22.19	10.83
2014-05-07	647	141	63	21.79	9.74
2014-05-08	709	156	65	22.00	9.17
2014-05-09	687	148	64	21.54	9.32
2014-05-10	69	22	20	31.88	28.99
2014-05-11	86	29	25	33.72	29.07
2014-05-12	1174	258	117	21.98	9.97
2014-05-13	807	193	92	23.92	11.40
2014-05-14	667	151	62	22.64	9.30

Result 7

Insights:

1. Weekly user engagement trends highlight active user patterns. Spikes may indicate successful feature launches or marketing campaigns.

2. Growth trends show the pace at which users join the platform. Sharp increases may point to effective onboarding strategies or viral events.

3. Retention metrics help identify how well the platform retains users after sign-up. Strong weekly retention suggests user satisfaction and engagement.

4. Device-level engagement provides insights into user preferences, helping prioritize mobile or desktop experiences.

5. Tracking email action rates evaluates the effectiveness of email campaigns. Low engagement may signal the need for improved content or targeting.

Results:

1. **Successfully executed SQL queries to investigate user engagement, growth, retention, and email activity.**
2. **Identified potential metric spikes linked to user actions and email interactions.**
3. **Delivered actionable insights for product and marketing teams.**

Hiring Process Analytics

Module - IV

Project Description:
This project focuses on analysing the hiring data to gain insights into the company's recruitment process. The main objectives are to explore hiring patterns, salary distributions, departmental and position tier analysis, and identify trends that can help improve future hiring strategies.

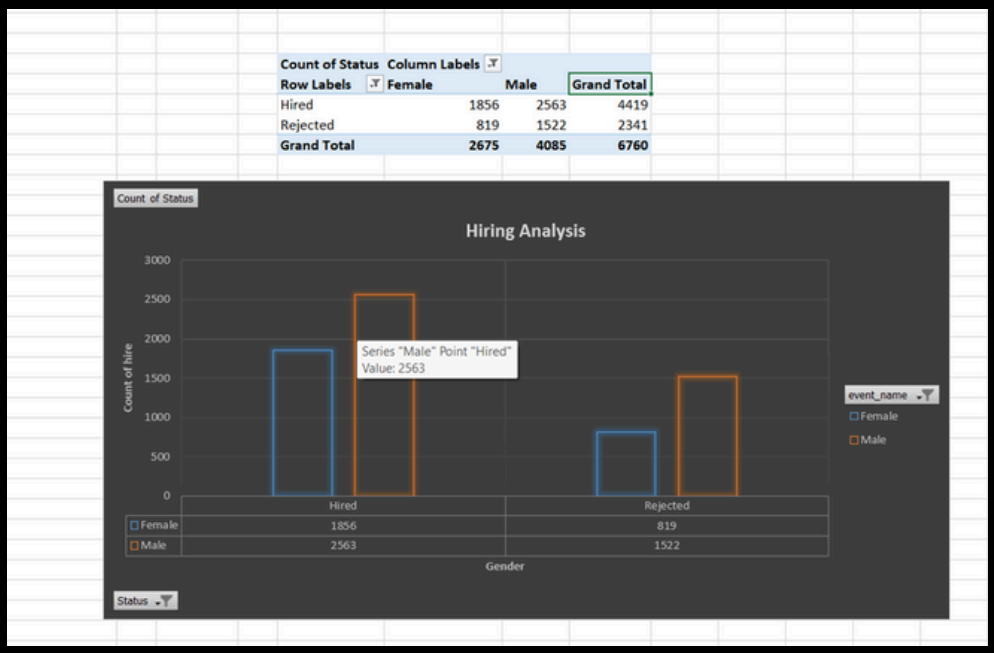
Approach:

1. **Data cleaning for handling missing values and outliers.**
2. **Data analysis using excel to perform the tasks.**
3. **Data visualisation to interpret and draw the insights.**

Tech-Stack Used:
Microsoft excel 2019 is used for statistical calculations, creating visualisations like bar graphs and pie charts, forming pivot tables and charts to interpret hiring data.

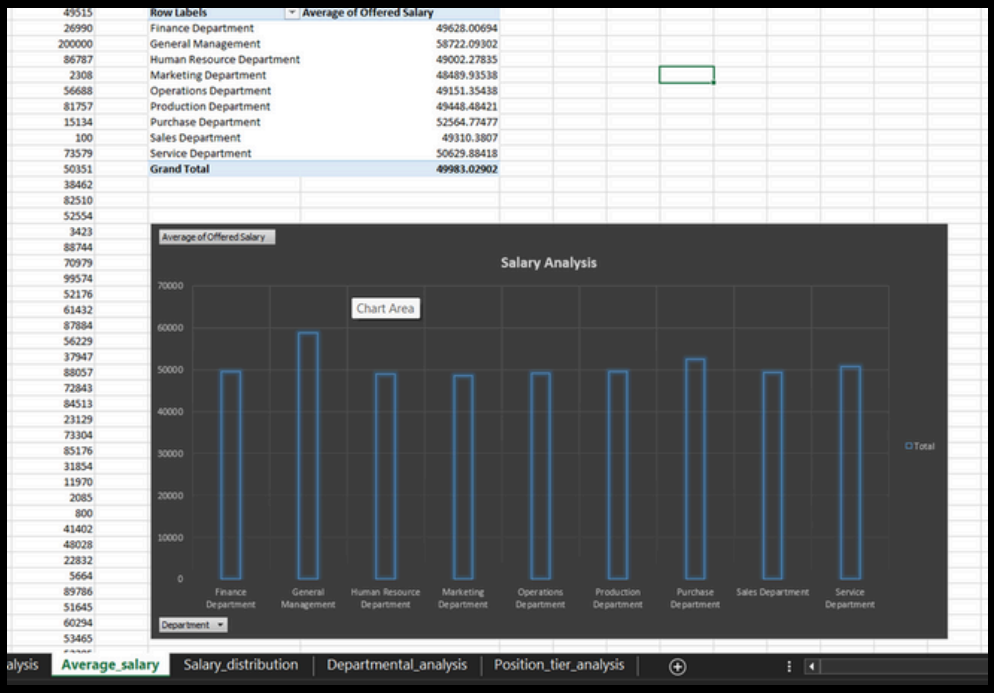
Insights:

1. **Hiring Analysis: The gender distribution analysis revealed the proportion of male and female hires, providing a clear picture of gender representation in the company.**



Count of Status	Column Labels		
Row Labels	Female	Male	Grand Total
Hired	1856	2563	4419
Rejected	819	1522	2341
Grand Total	2675	4085	6760

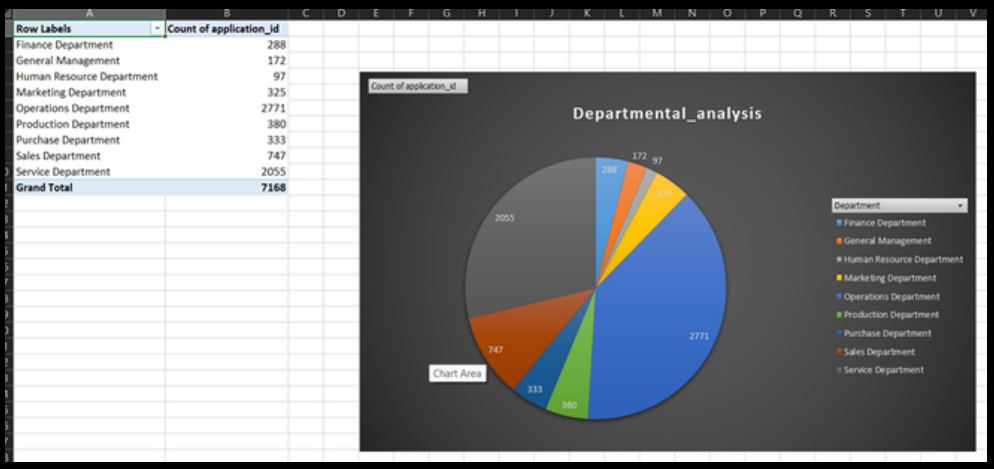
2. Salary Analysis: The average salary offered was calculated, and class intervals for salary distribution highlighted the spread and concentration of salaries across various ranges.



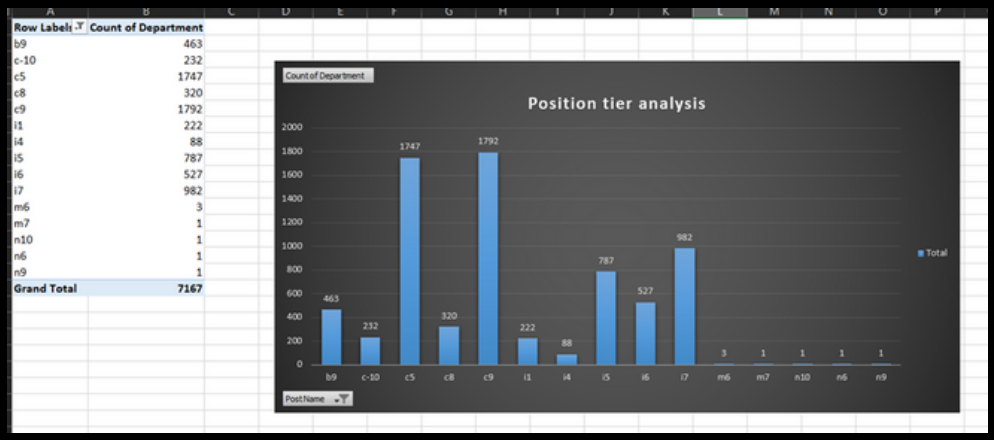
Row Labels	Average of Offered Salary
Finance Department	49628.00694
General Management	58722.09302
Human Resource Department	49002.27835
Marketing Department	48489.93538
Operations Department	49151.35438
Production Department	49448.48421
Purchase Department	52564.77477
Sales Department	49310.3807
Service Department	50629.88418
Grand Total	49983.02902

3. Salary Distribution: The salary distribution showed that most salaries fall within the 10,000–50,000 range, with fewer employees earning above 100,000. The distribution exhibited a right-skewed pattern, indicating limited high-tier positions compared to mid-level roles.

4. Departmental Analysis: Visual representations showed the distribution of employees across departments, identifying areas with higher or lower staffing.



5. Position Tier Analysis: Analysed the spread of job tiers, helping to understand the distribution of roles within the organization.

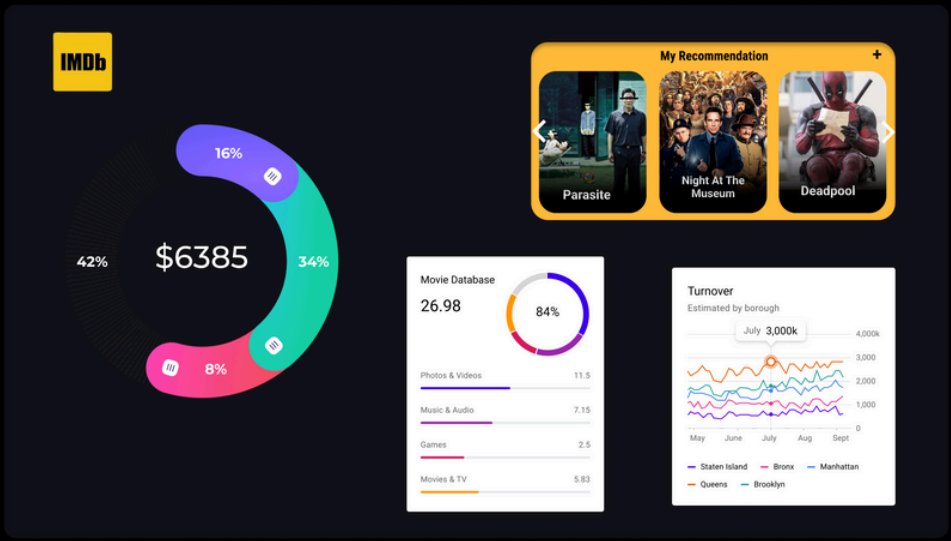


Row Labels	Count of Department
b9	463
c-10	232
c5	1747
c8	320
c9	1792
i1	222
i4	88
i5	787
i6	527
i7	982
m6	3
m7	1
n10	1
n6	1
n9	1
Grand Total	7167

Results:
Through this project, we gained a deeper understanding of the company's hiring patterns, salary distribution, and departmental composition. The insights derived can support data-driven decisions for more balanced hiring practices and salary structuring. This analysis contributes to improving recruitment strategies and identifying potential areas for optimization.

IMDb Movie Analysis

Module - V



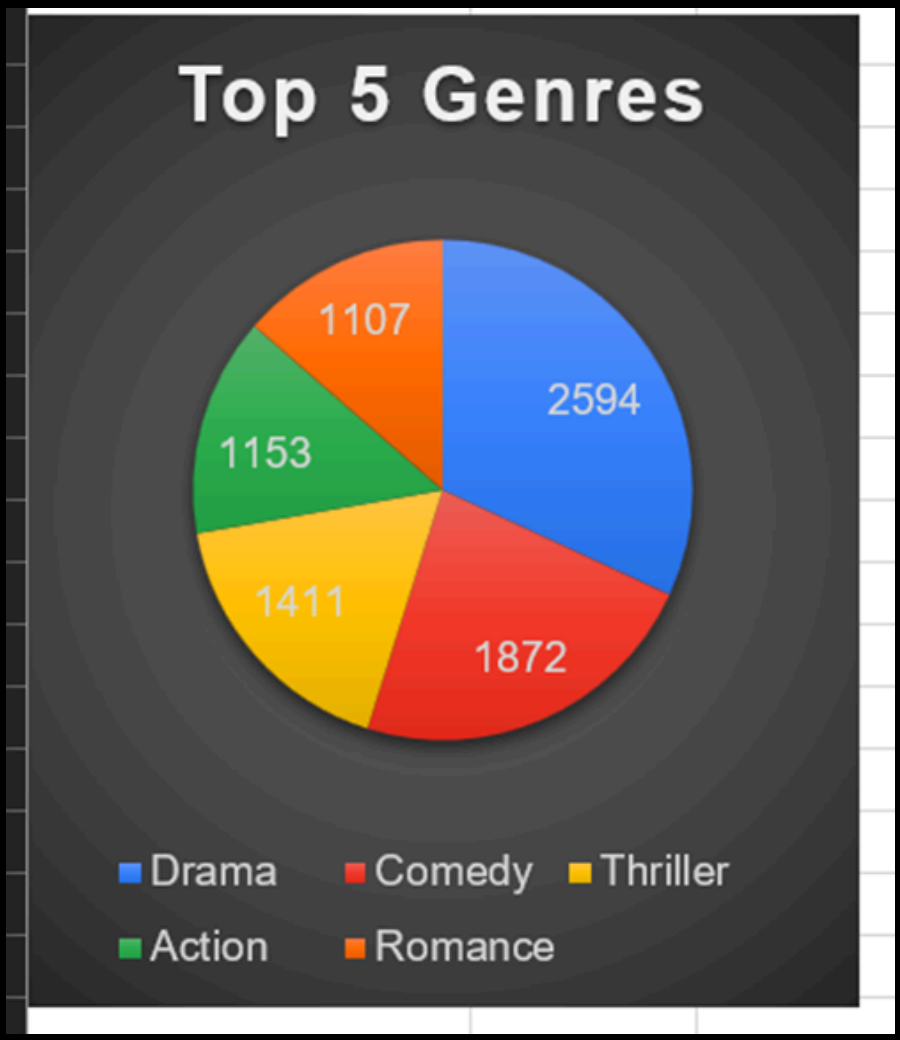
Project Description

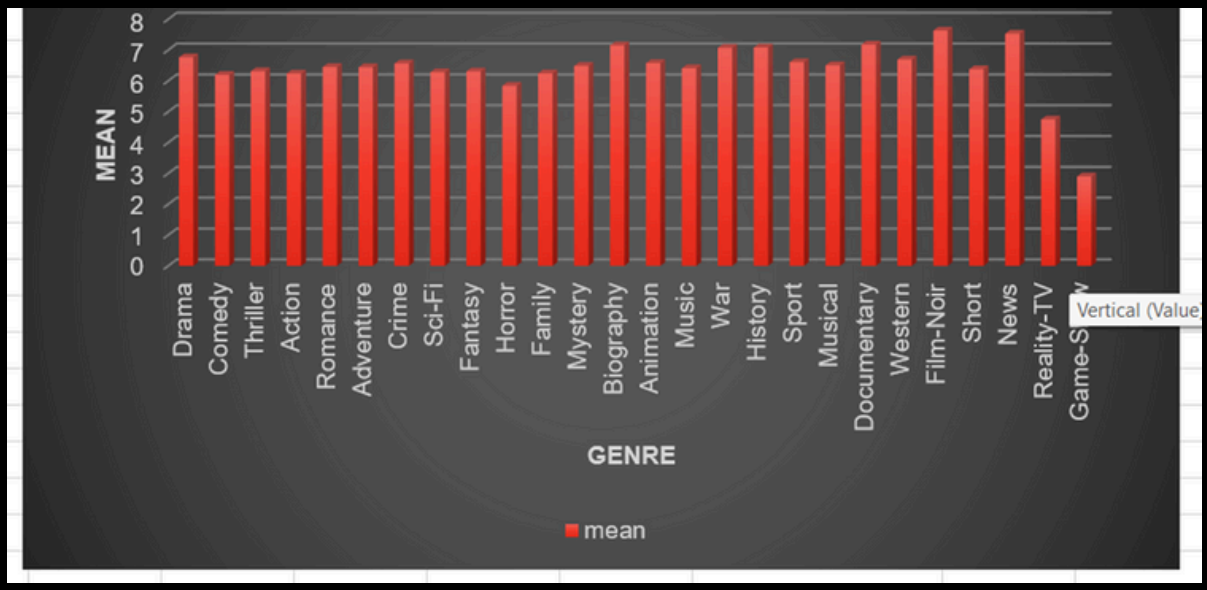
The IMDB Movie Analysis project aims to uncover insights about movies by analysing various factors such as genre, duration, language, director influence, and budget. The goal is to understand how these elements impact movie ratings and financial success, providing data-driven insights for filmmakers, producers, and enthusiasts.

Approach

1. Genre Analysis:

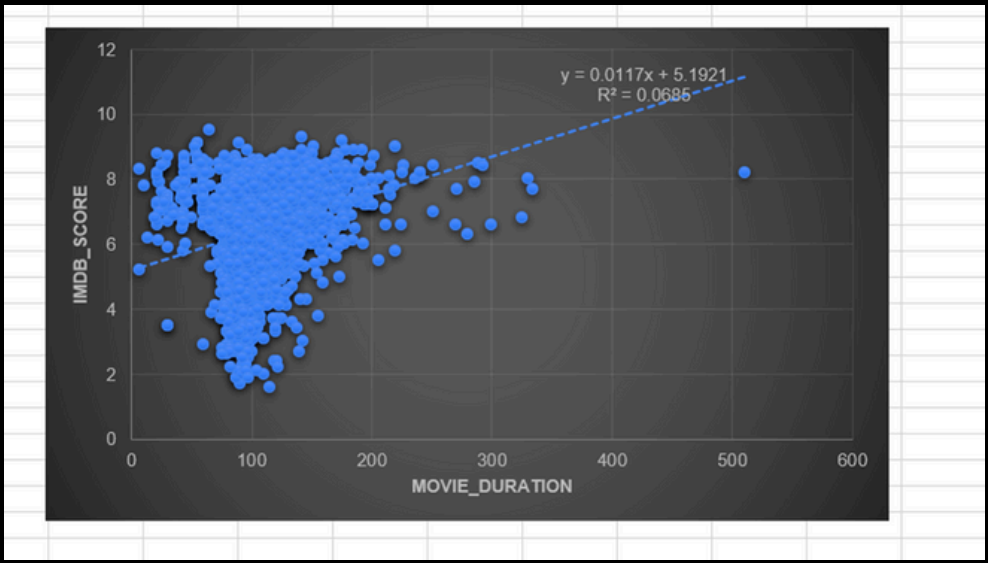
- **Calculated descriptive statistics (mean, median, mode) for IMDB scores across unique genres.**
- **Used Excel functions like AVERAGE, MEDIAN, and MODE to identify trends.**
- **Compared statistics to explore the relationship between genre and movie ratings.**


Top 5 Genres
1107
2594
1153
1411
1872
Drama
Comedy
Thriller
Action
Romance


MEAN
8
7
6
5
4
3
2
1
0
Drama
Comedy
Thriller
Action
Romance
Adventure
Crime
Sci-Fi
Fantasy
Horror
Family
Mystery
Biography
Animation
Music
War
History
Sport
Musical
Documentary
Western
Film-Noir
Short
News
Reality-TV
Vertical (Value
GENRE
mean

2. Duration Analysis:

- Cleaned the data by removing blanks in the movie duration column.
- Computed mean, median, and standard deviation for movie durations.
- Created a scatter plot in Excel to visualize the relationship between movie duration and IMDB scores.
- Added a trendline to assess the strength and direction of the correlation.

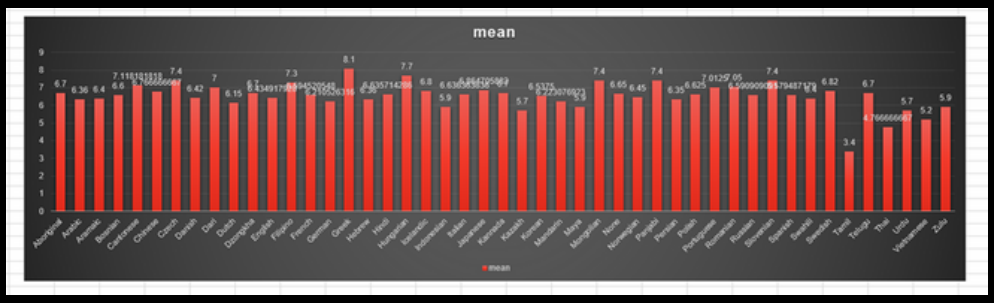


average	107.2011
median	103
std_dev	25.19744

3. Language Analysis:

- Created a unique language column.
- Used COUNTIF to count the number of movies for each language.
- Calculated mean, median, and standard deviation for IMDB scores for each language.
- Compared language-wise statistics to observe their impact on ratings.

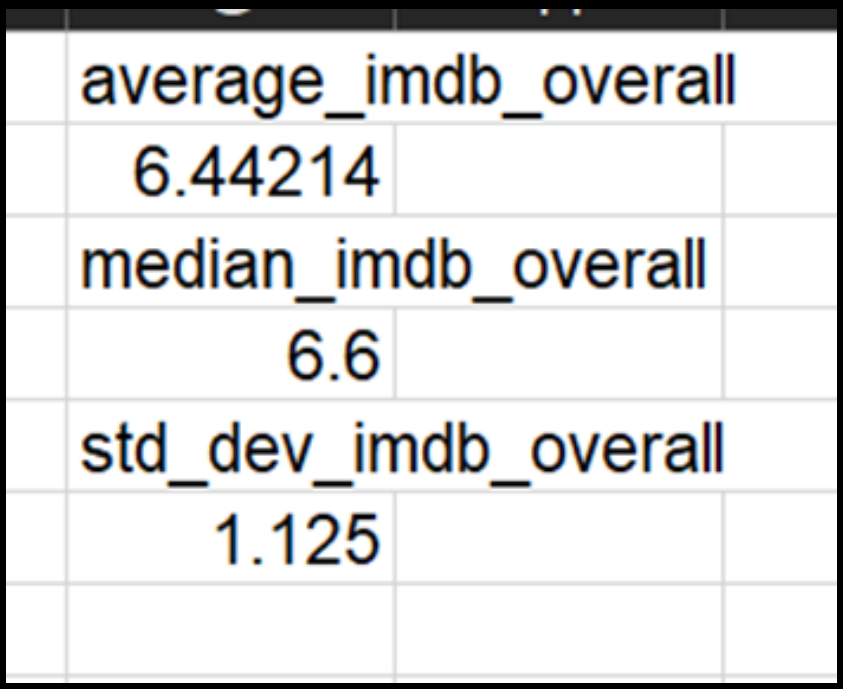
language (unique)	count	mean	median	std_dev
Aboriginal	2	6.7	6.7	0.2
Arabic	5	6.36	6.8	0.62482
Aramaic	1	6.4	6.4	0
Bosnian	1	6.6	6.6	0
Cantonese	11	7.11818	7.2	0.49326
Chinese	3	6.76667	6.9	1.06562
Czech	1	7.4	7.4	0
Danish	5	6.42	6.5	1.68214
Dari	2	7	7	1.4
Dutch	4	6.15	6.35	0.50249
Dzongkha	1	6.7	6.7	0
English	4691	6.43492	6.6	1.11767
Filipino	1	7.3	7.3	0
French	73	6.59452	6.9	1.27676
German	19	6.21053	6.7	1.51758
Greek	1	8.1	8.1	0
Hebrew	5	6.36	6.6	0.55353
Hindi	28	6.63571	6.75	1.02058
Hungarian	1	7.7	7.7	0



4. Director Analysis:

- Generated a list of unique directors.
- Computed the average IMDB score for each director using AVERAGEIF.
- Applied Excel's PERCENTILE function to identify top-performing directors.
- Compared their scores against the overall IMDB score distribution

director_name (unique)	average_imdb_scores	directors_based_on_scores
James Cameron	7.914285714	Top 10%
Gore Verbinski	6.985714286	Below Top 10%
Sam Mendes	7.5	Top 10%
Christopher Nolan	8.425	Top 10%
Doug Walker	7.1	Below Top 10%
Andrew Stanton	7.733333333	Top 10%
Sam Raimi	6.907692308	Below Top 10%
Nathan Greno	7.8	Top 10%
Joss Whedon	7.925	Top 10%
David Yates	7.05	Below Top 10%
Zack Snyder	7.175	Below Top 10%
Bryan Singer	7.2875	Below Top 10%
Marc Forster	7.15	Below Top 10%
Andrew Adamson	7.08	Below Top 10%
Rob Marshall	6.6	Below Top 10%
Barry Sonnenfeld	6.457142857	Below Top 10%
Peter Jackson	7.675	Top 10%
Marc Webb	7.133333333	Below Top 10%
Ridley Scott	7.070588235	Below Top 10%



5. Budget Analysis:

- Analysed the correlation between movie budgets and gross earnings using the CORREL function.
- Calculated profit margins (gross earnings - budget) for each movie.
- Used MAX to identify the top 10 most profitable movies

budget	gross	profit_margin	rank	movie_title
237000000	7.61E+08	523505847	1	Avatar
150000000	6.52E+08	502177271	2	Jurassic World
200000000	6.59E+08	458672302	3	Titanic
11000000	4.61E+08	449935665	4	Star Wars: Episode IV - A New Hope
10500000	4.35E+08	424449459	5	E.T. the Extra-Terrestrial
220000000	6.23E+08	403279547	6	The Avengers
45000000	4.23E+08	377783777	7	The Lion King
115000000	4.75E+08	359544677	8	Star Wars: Episode I - The Phantom Menace
185000000	5.33E+08	348316061	9	The Dark Knight
78000000	4.08E+08	329999255	10	The Hunger Games
58000000	3.63E+08	305024263	11	Deadpool
130000000	4.25E+08	294645577	12	The Hunger Games: Catching Fire
63000000	3.57E+08	293784000	13	Jurassic Park
76000000	3.68E+08	292049635	14	Despicable Me 2
58800000	3.5E+08	291323553	15	American Sniper
94000000	3.81E+08	286838870	16	Finding Nemo
150000000	4.36E+08	286471036	17	Shrek 2

movie_genre_analysis | movie_duration_analysis | language_analysis | director_analysis | budget_analysis

correlation_coefficient
0.101411

highest_profit_margin
5.24E+08

Tech-Stack Used

Microsoft Excel: For data cleaning, statistical analysis, visualization (scatter plots, trendlines), and correlation studies.
Functions Used: AVERAGE, MEDIAN, MODE, STDEV, COUNTIF, CORREL, MAX, PERCENTILE, AVERAGEIF.

Insights and Results

1. Genre Impact: Certain genres like Thriller and Adventure showed higher average IMDB scores, while others like Romance and Comedy had wider variability.
2. Duration Correlation: A slight positive correlation was observed between movie duration and IMDB scores — longer movies tended to have slightly better ratings.
3. Language Trends: English-dominated movies had the most entries, but other languages like French and Spanish showed promising average scores.
4. Director Influence: Top directors (as per PERCENTILE analysis) consistently produced high-rated movies, reinforcing their impact on a movie's success.
5. Budget vs. Profit: No strong correlation was found between high budgets and high ratings, but the top 10 most profitable movies had moderate budgets and massive box office earnings, emphasizing smart investments over extravagant spending.

Bank Loan Case Study

Module - VI

Project Description

The Bank Loan Case Study aims to analyse loan application data to identify key factors affecting loan defaults. The study involves handling missing data, detecting outliers, analysing data imbalance, performing statistical analysis, and identifying the most significant correlations that influence loan repayment behaviour.

Approach

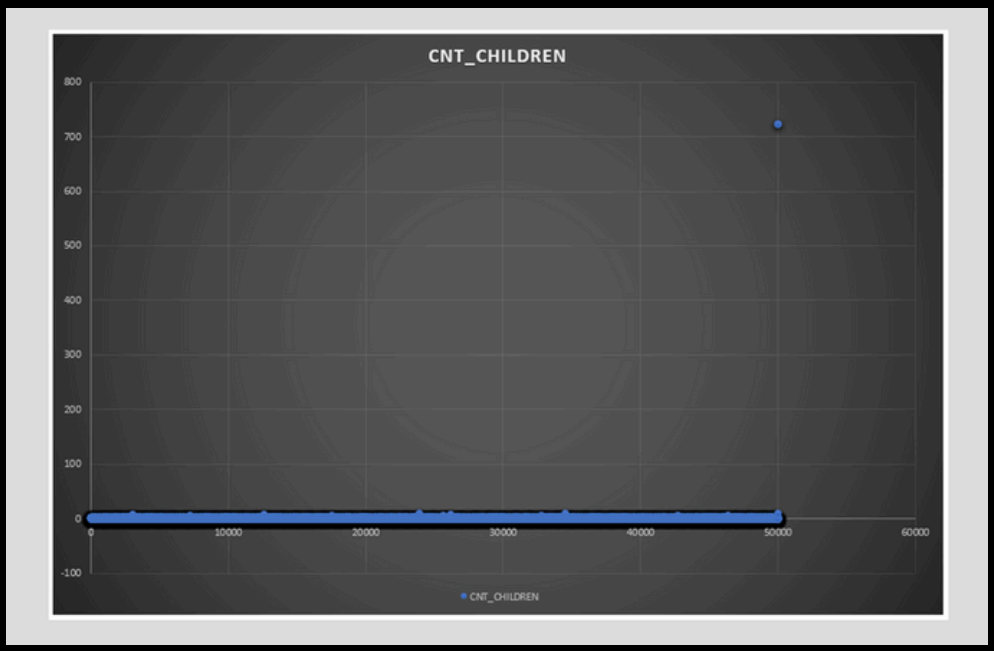
1. Handling Missing Data

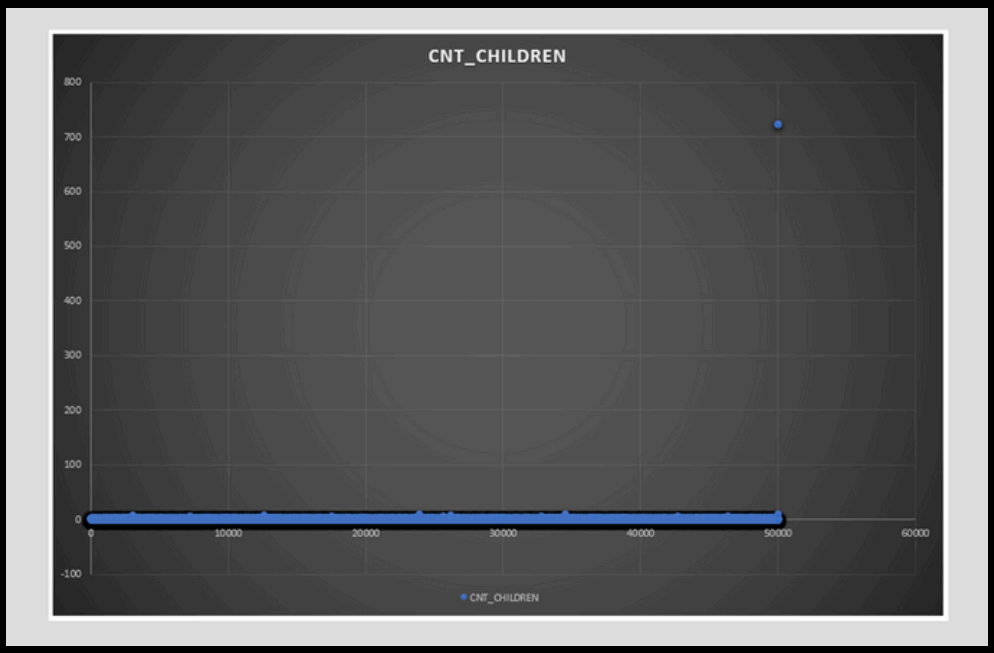
- **Identified missing values using COUNTIF function.**
- **Approach for handling missing values is explained.**
- **Created a bar chart to visualize missing values across variables.**

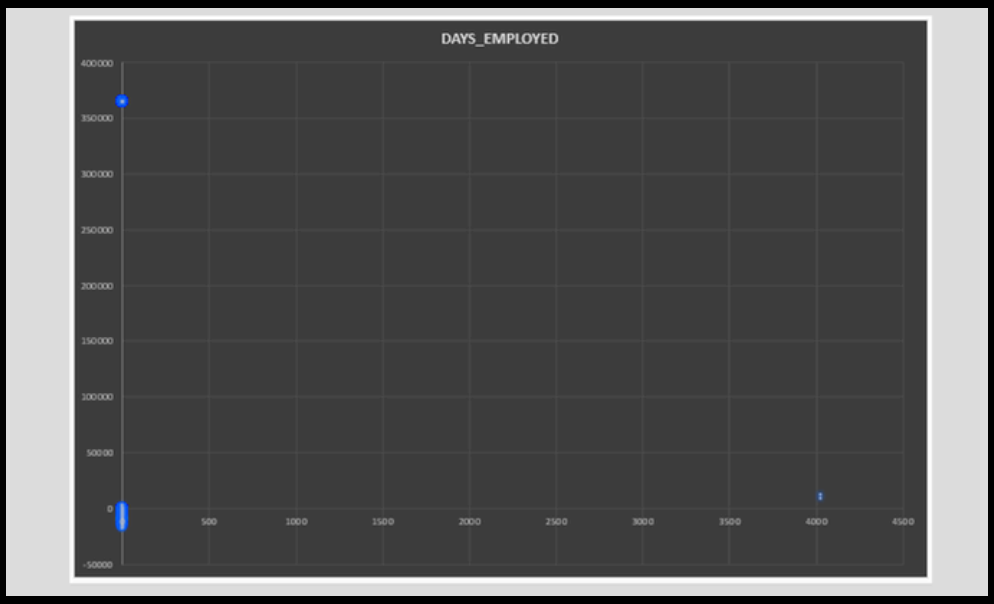
2. Detecting Outliers

- **Used QUARTILE and IQR methods to detect outliers in numerical**
- **variables like AMT_INCOME_TOTAL, DAYS_EMPLOYED and others**
- **Created scatter plots to visualize outlier distribution.**
- **Removed extreme outliers whereever necessary for accurate**
- **analysis.**

	no_of_outliers	max	validity_of_outliers	
)	0	157875	no outliers	
)	4026	1	validity_of_outliers	
)	723	11	not valid	no of children==11 not common
)	2295	117000000	not valid	max income value is too high
)	1063	4050000	valid	
)	1188	258025.5	valid	
)	2387	4050000	valid	
)	1329	0.072508	valid	
)	0	-7680	no outliers	
8	11712	365243	not valid	days employed cannot exceed person's life span
6	96	0	valid	
)	0	0	no outliers	

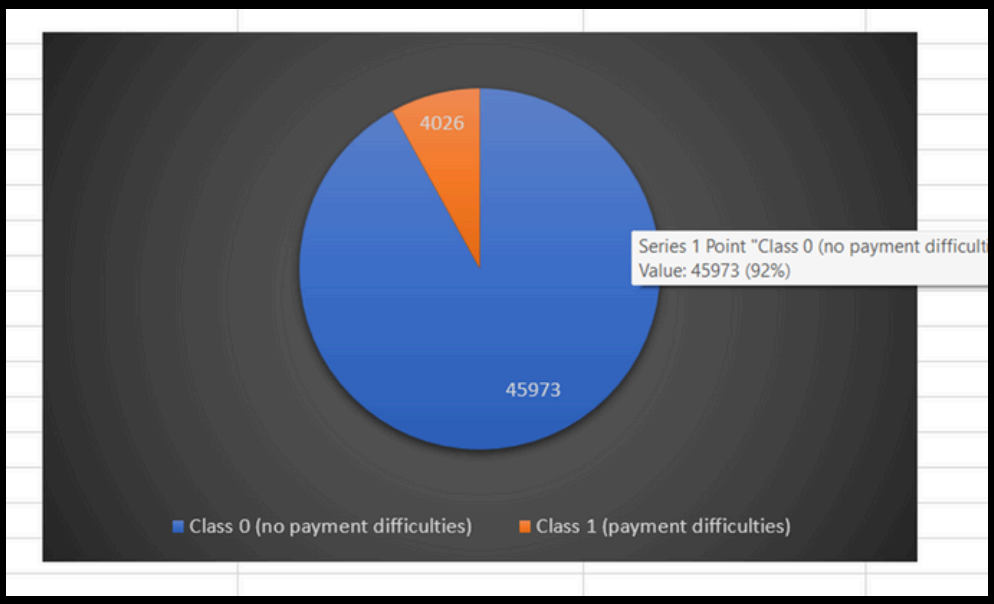






3. Data Imbalance Analysis

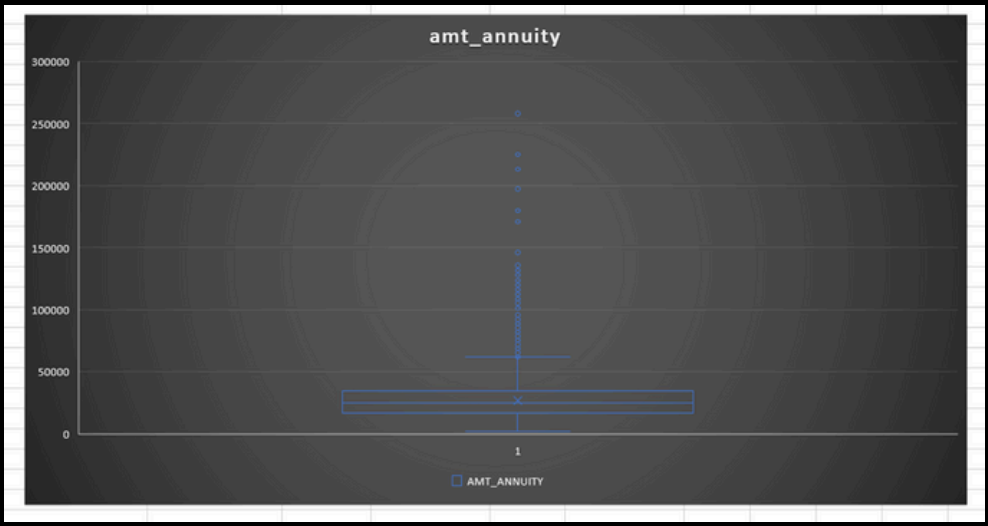
- Checked imbalance in the TARGET variable using COUNTIF.
- Calculated the proportion of TARGET = 1 (loan defaults) and
- TARGET = 0 (non-defaults).
- Created a pie chart to visualize data imbalance.

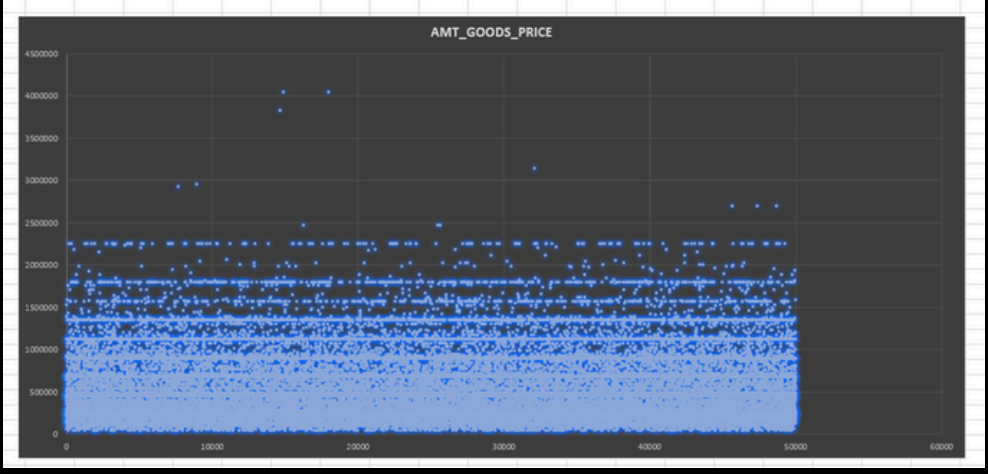


Ratio of data imbalance	
0.087573141	<<0.5
	highly imbalanced

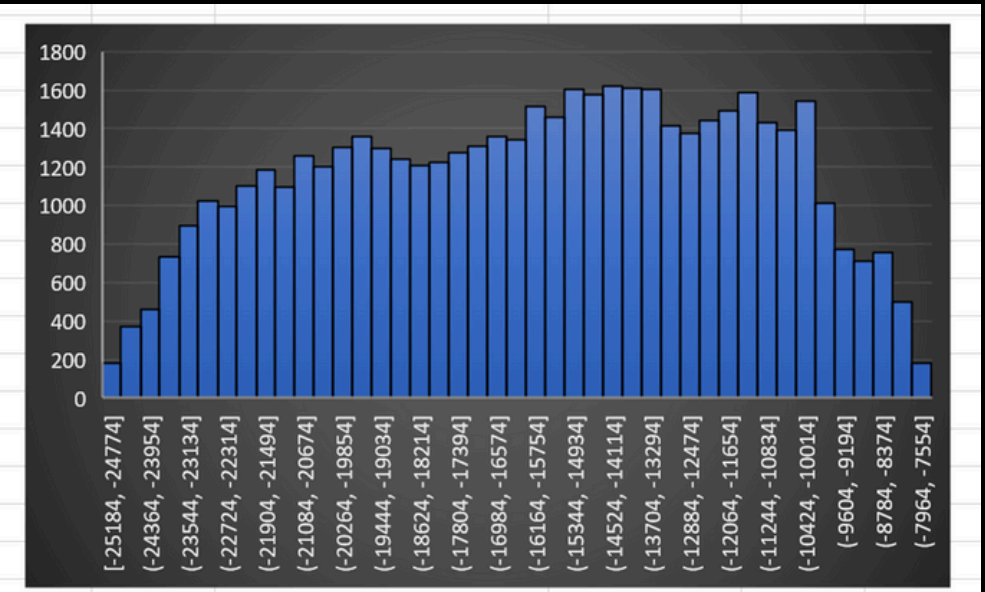
4. Univariate, Segmented Univariate, and Bivariate Analysis

- **Performed univariate analysis on key financial and demographic factors using AVERAGE, MEDIAN, and STDEV. Also, prepared histograms, box plots and scatter plot for different variables for visualization**

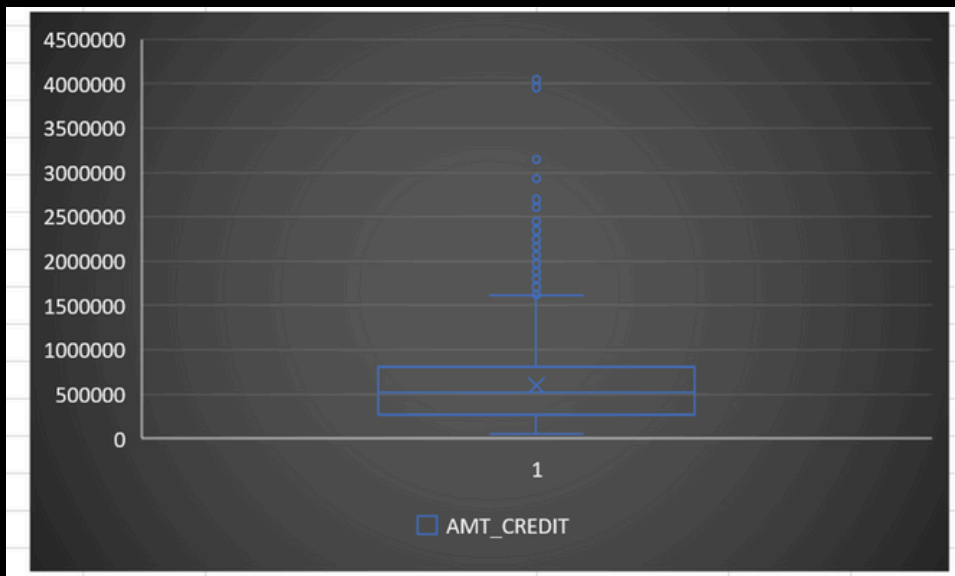




- **Conducted segmented univariate analysis by calculating the descriptive statistics based on Targett variable. Created charts for visualization.**



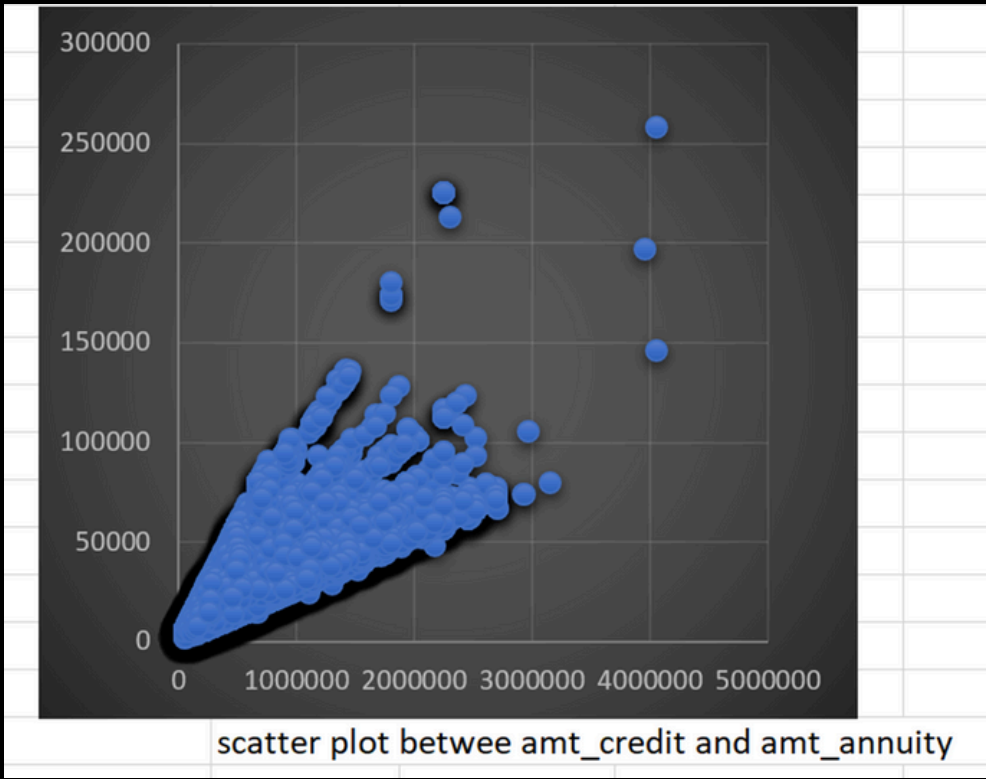
histogram for days_birth



box_plot for amt_credit

- **Performed bivariate analysis by creating a correlation matrix on major factors influencing the Target variable. Also created a scatter plot for better visualization.**

	amt_credit	amt_annuity	amt_goods_price	days_birth	days_registration
amt_credit	1	0.76949891	0.98694373	-0.059343	0.003448569
amt_annuity	0.7694989	1	0.774433947	0.007712	0.033218936
amt_goods_price	0.9869437	0.77443395	1	-0.057611	0.006101039
days_birth	-0.059343	0.00771224	-0.057610698	1	0.333632509
days_registration	0.0034486	0.03321894	0.006101039	0.333633	1



5. Correlation Analysis & Key Indicators of Default

- **Used the CORREL function to compute correlations for each**
- **segment.**
- **Identified top correlated features influencing loan defaults.**
- **Created heatmap to compare top correlations between both groups.**

	Target
TARGET	1
CNT_CHILDREN	0.987614361
CNT_FAM_MEMBERS	0.978237432
AMT_INCOME_TOTAL	-0.001252633
AMT_CREDIT	-0.009739591
AMT_ANNUITY	-0.011347341
AMT_GOODS_PRICE	-0.0096048
DAYS_BIRTH	0.024029408
DAYS_EMPLOYED	-0.002804753
DAYS_REGISTRATION	0.009463307
DAYS_ID_PUBLISH	0.0130521
DAYS_LAST_PHONE_CHANGE	0.008189177
FLAG_EMP_PHONE	0.499886927
FLAG_WORK_PHONE	0.999924249
FLAG_CONT_MOBILE	0.493690162
FLAG_PHONE	-0.004268107
FLAG_EMAIL	0.999857896
REGION_RATING_CLIENT	0.91717667

Tech Stack Used·
Microsoft Excel (Data cleaning, statistical functions, charts)
Data Visualization (Bar charts, Box plots, Scatter plots, Heatmaps)

Insights and Results
1. Missing Data: Significant missing values in some financial attributes required imputation.
2. Outliers: Extreme outliers were detected in CNT_CHILDREN, AMT_INCOME_TOTAL and DAYS_EMPLOYED.
3. Imbalance Analysis: The dataset was highly imbalanced, with a small proportion of TARGET = 1 cases.
4. Loan Default Factors: Key financial indicators, such as AMT_CREDIT, AMT_ANNUITY, and DAYS_CREDIT, showed strong correlations with loan defaults.
5. Segmented Correlations: The correlation between AMT_CREDIT and AMT ANNUITY was stronger in defaulters, indicating potential financial distress as a predictor.

Car Features Analysis

Module- VII

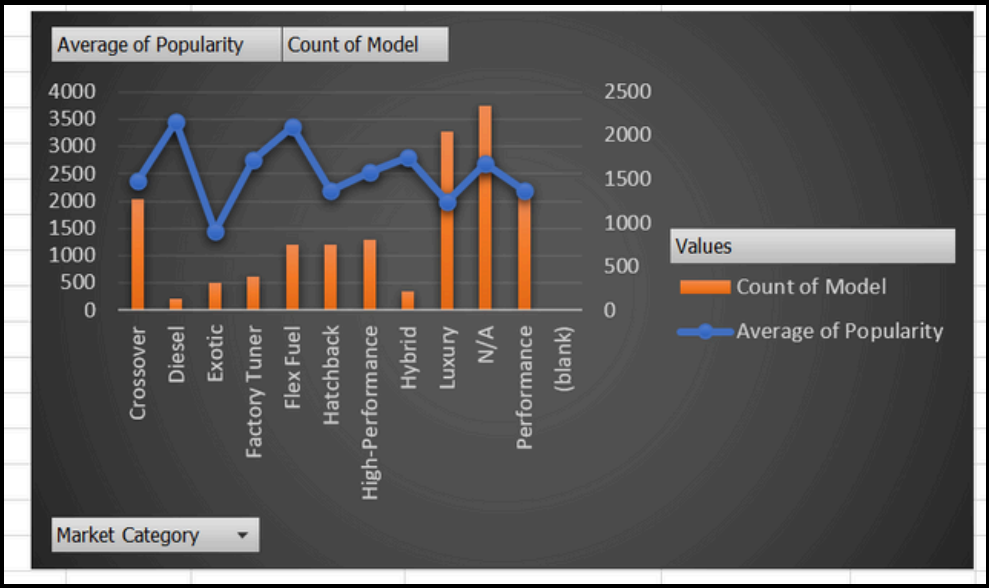
Project Description: This project focuses on analyzing a car dataset to extract meaningful insights related to pricing, fuel efficiency, and brand performance. The goal is to understand patterns in car prices, how features like engine size and body style affect market value, and how fuel efficiency varies across models.
The dataset used contains information on over 11,900 cars, including variables like Make, Model, Year, Engine HP, Transmission Type, MSRP, Fuel Type, MPG, and more.

Approach:
The analytical approach for this project was centered entirely around Excel Pivot Tables and data visualizations. Pivot tables were used to summarize and analyze the dataset by aggregating values such as average MSRP, total MSRP, fuel efficiency, and other metrics across various categories like car brand, body style, transmission type, and model year. This method allowed for dynamic data exploration and quick extraction of meaningful patterns without the need for complex formulas or additional functions.

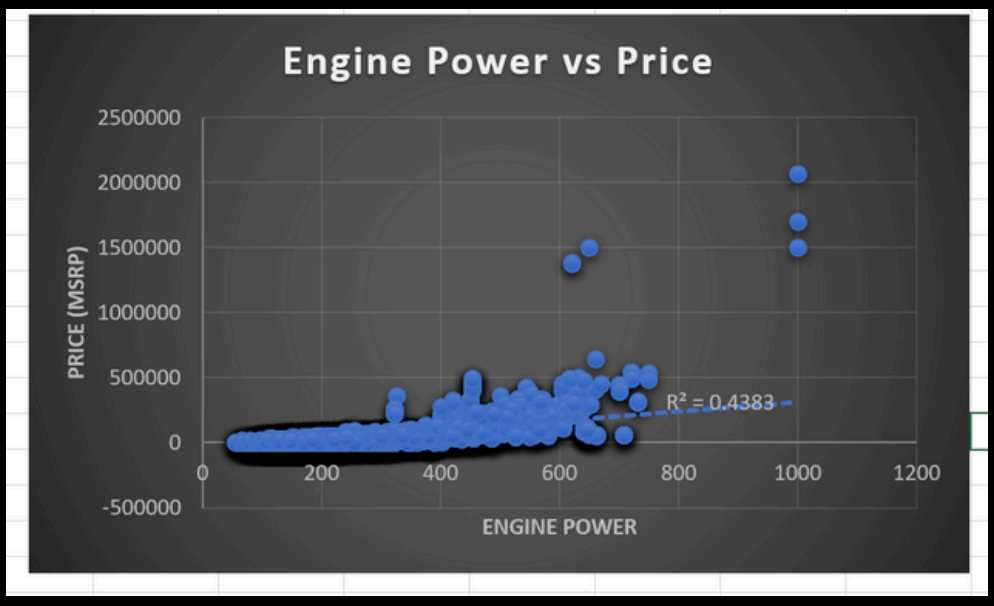
We utilized a variety of chart types—such as stacked column charts, clustered column charts, scatter plots, and line graphs—to visualize relationships and trends in the data. Each chart was made interactive using slicers and filters, enabling users to drill down into specific aspects of the dataset easily. The combination of pivot tables and interactive charts formed the core analytical technique throughout the project, providing a user-friendly way to answer business questions and generate actionable insights.

Tech stack used:
Microsoft Excel for the entire workflow: data cleaning, pivot tables, and visualization which provides:
1. Highly accessible and widely used.
2. Offers dynamic features like slicers, filters, and interactive dashboards. Excel's Data Analysis Toolpak for correlation coefficient.

Insights and Results:



Above figure shows the relationship between different market categories with that of popularity of the model and the count of model. Popularity peaks for diesel market category whereas highest count of model turns out to be unknown followed by luxury market category.



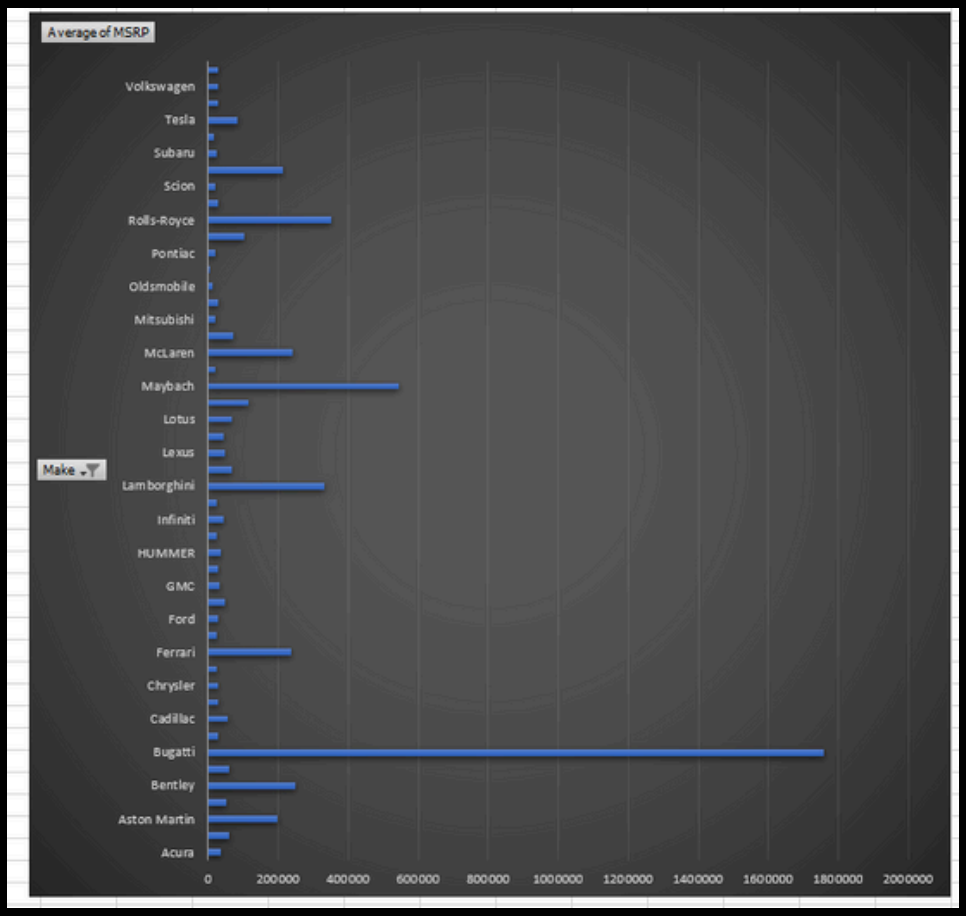
Above figure shows the scatter plot between engine power and MSRP (price). Relationship is found to be linearly increasing.

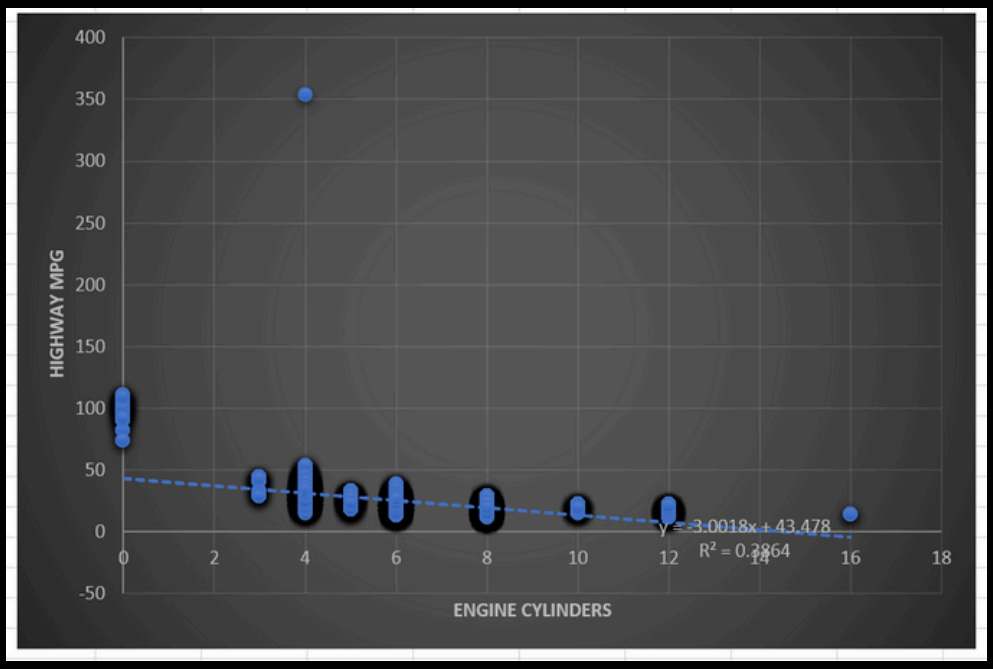
SUMMARY OUTPUT	
Regression Statistics	
Multiple R	0.674145591
R Square	0.454472278
Adjusted R Square	0.454151541
Standard Error	44409.54215
Observations	11914

The above figure shows the regression statistics.

- Multiple R value suggests a moderate positive relationship between predictors and target. Adjusted R square close to R square indicates a good number of relevant predictors. Standard error indicates error in predictions.
- Observations refer to sample size which in this case is a solid amount.
- Model's p-value comes out to be near zero from further analysis, which is highly significant and f-score turns out to be very high, indicating overall regression model is statistically significant.
- The only insignificant predictor: city MPG.

The figure below shows the relationship between highway mpg with engine cylinders which turns out to be linearly decreasing, i.e. inversely proportional. The correlation coefficient is close to -1 suggesting the same.

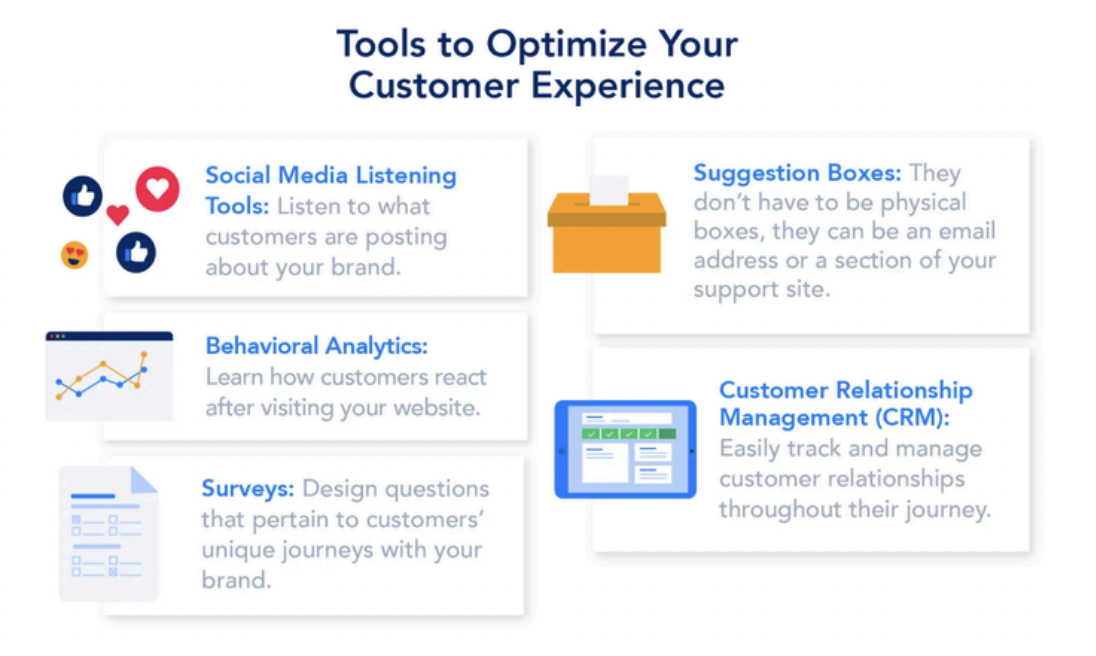




The figure above shows average of MSRP based on the brand (make). Highest price goes to the Bugatti model.

ABC Call Volume Trend Analysis

Module - VIII



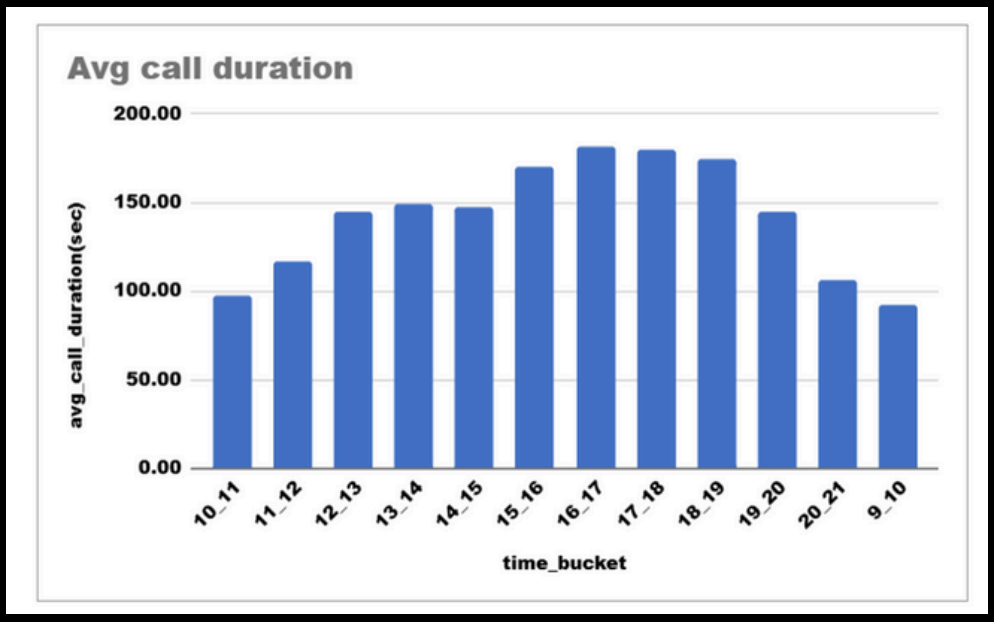
Project Description: This project focuses on Customer Experience (CX) analytics for ABC Insurance Company, specifically analyzing inbound call data spanning 23 days. The goal is to derive insights that help optimize call handling, improve customer satisfaction, and support strategic manpower planning. Key call-related metrics such as duration, volume, and abandon rate were analyzed across different time buckets.

Approach: We performed an in-depth data analysis using Microsoft Excel to clean and structure the dataset. Calculations and visualizations were developed to determine average call durations, call volumes by time bucket, and agent requirements to reduce the abandon rate. We applied time-based aggregation, pivot tables, bar charts, and manpower efficiency assumptions to generate actionable insights and forecasts.

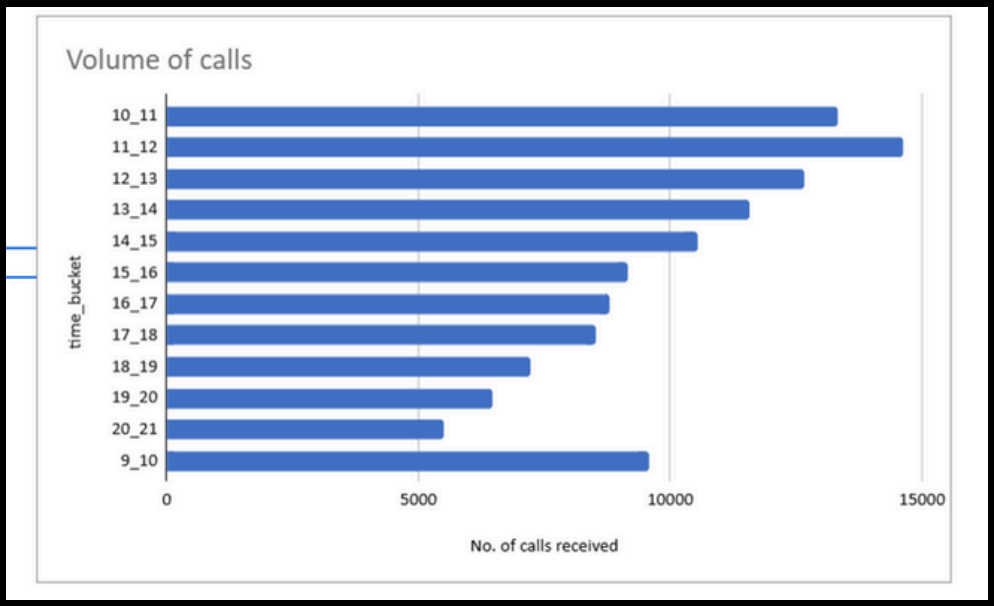
Tech Stack Used: Microsoft Excel: Used for time-series analysis, statistical calculations, and data visualization.

Insights:

1. Average Call Duration: Average duration was calculated for each time bucket (e.g., 9-10 am, 10-11 am, etc.). Higher average durations were noted in mid-morning and late afternoon slots, indicating possible complexity in customer queries during those periods.

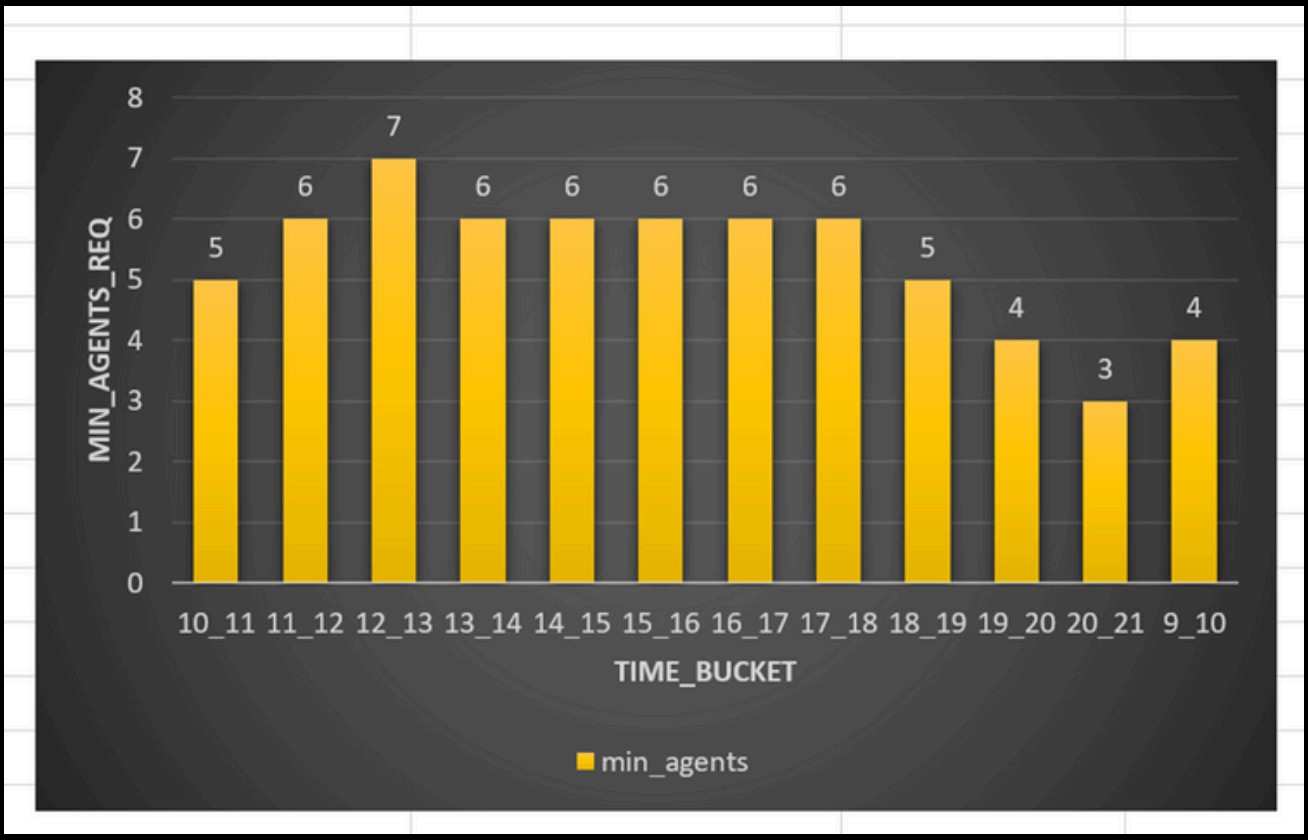


2. Call Volume Analysis: A time-bucketed bar chart showed peak call volumes between 10 am to 12 pm and again from 9 pm to 10 pm. These peaks suggest the need for focused manpower allocation during these hours.



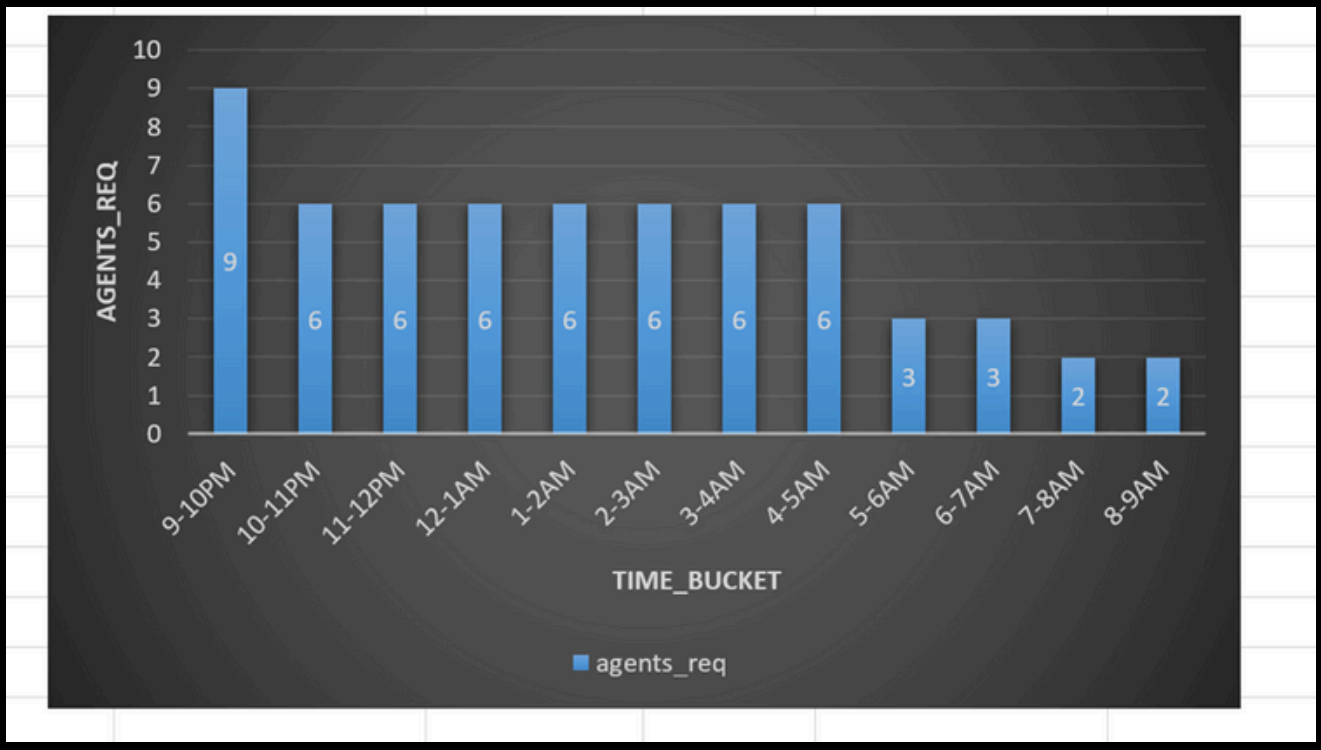
3. Manpower Planning: Based on the existing 30% abandon rate and target of 10%, the number of agents per time bucket was recalculated. The number of agents was proportionally scaled to ensure at least 90 out of 100 calls are answered during peak hours.

time_bucke	Sum of Call_Seconds (s)	avg_daily_duration	req_duration	min_agents
10_11	1297006	56391.56522	72503.44099	5
11_12	1708079	74264.30435	95482.67702	6
12_13	1831061	79611.34783	102357.4472	7
13_14	1728843	75167.08696	96643.39752	6
14_15	1552143	67484.47826	86765.75776	6
15_16	1556085	67655.86957	86986.11801	6
16_17	1594489	69325.6087	89132.92547	6
17_18	1533769	66685.6087	85738.63975	6
18_19	1261762	54859.21739	70533.2795	5
19_20	934437	40627.69565	52235.6087	4
20_21	583250	25358.69565	32604.03727	3
9_10	882195	38356.30435	49315.24845	4
Grand Total	**16463119**	715787.7826	920298.5776	57



4. Night Shift Planning: Assuming night calls are 30% of day calls, the call distribution was modeled across 9 pm to 9 am in hourly buckets. A manpower plan was proposed for night shifts to meet the same 10% maximum abandon rate, considering operational constraints.

time_bucket	%calls(assumed)	calls/night	call_sec	call_sec/day	agents_req
9-10PM	15	23085	3231900	140517.3913	9
10-11PM	10	15390	2154600	93678.26087	6
11-12PM	10	15390	2154600	93678.26087	6
12-1AM	10	15390	2154600	93678.26087	6
1-2AM	10	15390	2154600	93678.26087	6
2-3AM	10	15390	2154600	93678.26087	6
3-4AM	10	15390	2154600	93678.26087	6
4-5AM	10	15390	2154600	93678.26087	6
5-6AM	5	7695	1077300	46839.13043	3
6-7AM	5	7695	1077300	46839.13043	3
7-8AM	3	4617	646380	28103.47826	2
8-9AM	2	3078	430920	18735.65217	2
				total_agents_req	61



Results: The project resulted in a detailed hourly manpower allocation plan aimed at reducing call abandon rates to 10% during both day and night operations. The analysis also revealed high-demand periods, enabling more efficient resource scheduling. This contributes to improved customer satisfaction and operational efficiency for ABC Insurance Company's inbound support team.

Thank You

GiHub: https://github.com/sneha054

LinkedIn: www.linkedin.com/in/sneha-katole

Gmail: snehakatole02@gmail.com