Assignment 3

ML Group Project

24/05/2024

Group 7

Student Last Name	Student First Name	Student ID	Group Allocation
Dangol	Astha	25173651	Student A
Rejon	Rakibul Hassan	24880419	Student B
Saleh	Shaqran Bin	25010238	Student C
Amjad	Fahad	25116928	Student D

36106 - Machine Learning Algorithms and Applications
Master of Data Science and Innovation
University of Technology of Sydney

Table of Contents

1. Executive Summary

This project highlights the transformative impact of data science in banking. We, the bank's first data scientists, leveraged 3 years of transactional data to develop Machine Learning solutions for key business needs. These solutions aimed to improve customer experience, marketing efficiency, and fraud detection.

The core problem statements we addressed were:

- **Financial Empowerment for Customers:** We strived to equip customers with better tools for financial management by predicting their total monthly spending. This would allow them to approach budgeting and planning with greater ease.
- **Fraud Detection and Risk Mitigation for Compliance:** Our focus on identifying fraudulent transactions directly supported the Compliance Team's efforts to mitigate financial risks for the bank.
- **Targeted Marketing Strategies:** By segmenting customer bases based on spending behaviors, we aimed to facilitate the development and execution of highly targeted marketing campaigns for the Marketing Team. This approach would not only increase engagement but also boost conversion rates.
- **Proactive Customer Support:** Early detection of abnormal spending patterns was a key objective. By building an anomaly detection system, we empowered the Customer Support Team to proactively reach out to customers who might require assistance.

The project's outcomes were as follows:

- **Financial Budgeting Model:** A robust regression model was developed, enabling accurate predictions of individual customer spending for the following month. This directly translated to improved financial planning capabilities for the customer base.
- **Fraud Detection Model:** We created a classification model with a high degree of effectiveness in identifying potentially fraudulent transactions. This significantly enhanced the bank's overall security posture and fraud prevention capabilities.
- **Customer Segmentation:** By implementing clustering algorithms, we successfully grouped customers based on their spending behaviors. This provided the Marketing Team with valuable insights for designing and executing personalized and highly effective marketing campaigns.
- **Anomaly Detection System:** The developed system served as a vigilant watchdog, flagging unusual spending patterns. This enabled the Customer Support Team to intervene promptly and provide timely assistance to customers in need.

2. Business Understanding

a. Business Use Cases

Business Case 01

Helping customers to better budget their finances by predicting their total spending amount for the next month - Regression

The prediction of total spending amount for the next month can be applied across various sectors.

- In personal finance management, it helps customers to plan budgets by avoiding overspending and tracking expenses more efficiently.
- Banking and financial services can also leverage these predictions to offer personalized advice and improve credit risk assessments.
- Similarly, Retail and e-commerce businesses benefit by tailoring promotions and optimizing inventory management based on spending patterns.

Implementing a predictive spending model can face several challenges.

- An incomplete or missing data can significantly impact the model's accuracy.
- The variability in individual spending behavior adds complexity to achieving precise predictions.
- Additionally, gaining customer trust in these predictions is most crucial, as customers need to feel confident in the accuracy and privacy of the data being used.

Despite these challenges, there are significant opportunities.

- Enhancing customer experience by providing proactive financial management tools can empower users and foster loyalty.
- Businesses can gain valuable data-driven insights, leading to better decision-making.
- Likewise, offering personalized services based on spending predictions can differentiate businesses in competitive markets and improve customer retention.

Business Case 02

The project aims to improve the bank's capacity to forecast fraudulent transactions and comprehend consumer behavior by conducting thorough analysis of transaction details and personal information. This predictive model fulfills various essential roles in the bank's operations, including enhancing fraud detection systems, optimizing client profile, and enhancing marketing strategies.

Enhancing efficiency in banking operations using predictive analytics:

The approach is mostly utilized for targeted fraud detection. By categorizing transactions according to their probability of being fraudulent, the bank can concentrate its resources on transactions with a high risk of fraud, thus enhancing the effectiveness of its fraud prevention procedures. By employing this focused strategy, the bank may manage resources with greater efficiency, thereby swiftly identifying and mitigating potential instances of fraud. The model additionally offers important insights about customer expenditure habits and preferences, allowing the bank to customize its products and services to more effectively fulfill customer requirements.

Rationale for the Project:

The rationale behind using a machine learning-based model arises from the necessity to bolster security in financial transactions, optimize the allocation of resources, and tailor consumer interactions. Machine learning is well-suited for this role due to its ability to evaluate extensive amounts of data and identify patterns that may not be readily apparent using conventional approaches. This, in turn, facilitates data-driven decision-making in various financial operations.

Business Case 03:

The project explores the application of clustering techniques to enhance the bank's marketing strategies. By leveraging customer transaction data and personal information, the proposed approach involves segmenting the customer base into distinct clusters based on their spending patterns.

This customer segmentation allows for the development of targeted marketing campaigns. Each campaign can be tailored to address the specific needs and preferences of a particular customer segment, leading to a significant increase in the effectiveness of marketing efforts. Furthermore, insights achieved from customer behavior analysis can improve business development, ensuring a more strategic alignment with customer expectations

The project advocates for the adoption of a machine learning-based clustering model for several key reasons. Machine learning algorithms excel at processing large volumes of customer data and uncovering hidden patterns within that data through unsupervised learning. This allows for a more nuanced understanding of customer behavior and facilitates data-driven decision-making in both marketing and customer relationship management.

Machine learning clustering methods offer a multifaceted approach to marketing. By segmenting customers based on shared characteristics, clustering allows for targeted marketing campaigns that resonate with specific groups. This tailored approach enhances customer engagement and satisfaction while optimizing resource allocation and maximizing return on investment. Additionally, insights gleaned from cluster analysis inform product and service development, ensuring a more customer-centric strategy.

Business Case 04:

Anomaly detection machine learning algorithms such as Isolation Forest, Local Outlier Factor (LOF), and K-means clustering were utilized in the project to detect anomalies and flag potentially fraudulent transactions for banking transactions. Both numerical and categorical features of the combined customer and transactions dataset were implemented in order to understand key variables which would assist in these anomaly detection models. Fraud ('is_fraud') as the target variable has been utilized for these models and then further evaluation metrics have been used to find the best model approach.

It is critical for financial institutions and their stakeholders that fraud detection or anomaly detection in the transactions is given significance. This is to maintain integrity and trust of the stakeholders in their financial transactions oversighted by organizations and safeguarding of assets.

The stakeholders and their requirements will also be investigated with regards to implementation of these anomaly detection algorithms.

The management of the bank would require an effective fraud detection system to safeguard the bank's financial assets. This is important to seek solutions that minimize false positives to avoid inconveniencing legitimate customers.

The customers expect a secure banking environment where their transactions are protected from fraudulent activities. They would prefer minimal disruptions to their banking experience while ensuring the highest level of security.

The authorities that look after regulations are also essential as stakeholders. They need to ensure regulations related to fraud detection and prevention to maintain the integrity of the financial system are working effectively.

The risk management department requires accurate and timely identification of fraudulent transactions to mitigate financial risks associated with fraud. The work with insights into emerging fraud patterns to proactively enhance fraud prevention measures.

Key Objectives for Business Case 01:

The primary objective of the project is to **develop a machine learning regression model** that predicts the total spending amount for the next month.

Key stakeholders for this project include **customers, banks, and marketing and customer service teams.**

- **Customers** - They need accurate spending predictions to help manage their finances better and make informed decisions.
- **Banks and Financial Institutions** - They want to offer personalized financial advice and products, improve customer satisfaction, and understand customer behavior for better risk management and marketing.
- **Marketing and Customer Service Teams** - They use insights from the model to tailor marketing strategies, improve customer engagement, and provide better support based on spending patterns.

The project addresses these requirements by developing machine learning models trained on historical spending data to accurately predict future spending amount for next month. By accurately forecasting spending behavior, stakeholders can better plan and build strategies, leading to improved budgeting and financial decision-making.

Key Objectives for Business Case 02:

The main objective of this project is to create a predictive model that can effectively detect fraudulent transactions and offer valuable insights into customer behavior. The initiative aims to achieve three main objectives: boost security measures, optimize marketing strategies, and improve the overall consumer experience.

Key individuals or groups with a vested interest in a project, known as stakeholders, and the specific needs and expectations they have for the project.

The primary actors involved in this project comprise:

- **Fraud Detection Team:** Seeks techniques to enhance the identification and prevention of fraudulent transactions, resulting in less financial losses.
- **The Marketing Team** requires insights to enhance their ability to target campaigns and promotions effectively, hence improving conversion rates and optimizing marketing expenditure.

- **The Customer Service** Team aims to customize interactions with clients by utilizing comprehensive profiles and analyzing behavior trends.
- **Senior Management:** Seeks to achieve total corporate expansion by enhancing security measures, enhancing client loyalty, and optimizing the allocation of resources.
- **Customers anticipate** receiving customized services and ensuring that their financial transactions are both secure and tailored to their preferences and past activities.

Meeting the needs and expectations of stakeholders

The project seeks to fulfill these goals by employing machine learning techniques in the following manners:

Dear Fraud Detection Team, the model accurately detects transactions that have a high probability of being fraudulent. This allows the team to concentrate their efforts on these transactions and take preventive measures with more efficiency.

Regarding the Marketing Team: The model assists the marketing team in properly allocating their money and improving return on investment (ROI) by categorizing clients according to their transaction patterns. This enables the team to create tailored tactics and individualized communications.

The Customer Service Team benefits from predictive insights on customer behavior, which enables them to personalize their interactions and improve customer happiness and loyalty.

Senior management may enhance strategic decision-making by incorporating predictive analytics. This integration enables more data-driven and efficient operational approaches, resulting in improved resource allocation, increased customer happiness, and enhanced security.

Customers benefit from the implementation of predictive modeling, which guarantees secure transactions and tailored services that are pertinent to their individual requirements and past engagements. This enhances their overall experience and contentment with the bank.

Ultimately, the utilization of machine learning not only meets the distinct requirements of different parties involved but also propels the bank towards more effective and customer-focused operational methods, guaranteeing heightened security and tailored banking experiences.

Key Objectives for Business Case 03:

The project aims to enhance targeting in marketing campaigns, increase customer engagement with personalized messages, optimize resource allocation, and inform product and service development through insights gained from clustering customer data and segmenting customer groups.

The project addresses the needs of various stakeholders within the organization:

- Marketing Team: Gain access to detailed customer segments, enabling them to develop targeted marketing campaigns that resonate with each segment.
- Sales Team: Focusing efforts on the most profitable areas.
- Product Development Team: The product development team can make data-driven decisions about product and business development.
- Senior Management: Improve business performance and return on investment (ROI) with the help of this project through the optimization of marketing campaigns, sales efforts, and product development initiatives.

The project will leverage machine learning clustering algorithms to achieve customer segmentation. These algorithms will analyze customer transaction details and personal information to group customers with similar spending behaviors. This data-driven segmentation will provide valuable insights to inform marketing strategies, resource allocation, and product development roadmaps.

Key Objectives for Business Case 04:

The key objectives consisted of several aspects such as reducing financial losses due to fraudulent transactions by accurately identifying and flagging suspicious activities.

It is critical to enhance, build and maintain customer trust by ensuring the security and integrity of banking transactions. Moreover, compliance with regulations is also critical for fraud detection and prevention in financial institutions.

The objectives also consisted of streamlining fraud detection processes to improve operational efficiency and reduce manual effort. It is essential for banks to implement adaptability features. This helps to develop a scalable and adaptable fraud detection system capable of evolving with emerging fraud patterns and trends.

3. Data Understanding

Sources of Data and Methods of Collection:

The dataset used for the project was acquired from:

https://drive.google.com/file/d/1ipwbg0VClq9OGYP-B9wnFegKxPneMFug/view?usp=drive_link

The dataset consists of transaction records and personal information of bank clients, obtained from the bank's internal systems. The bank's transaction processing system automatically records transaction data, including credit card number, account number, transaction number, Unix timestamp, category, amount, fraud status, merchant name, and merchant geographic coordinates. Customer data is obtained during the process of registration and updates, which can be done through online forms, in-person applications, or customer service interactions.

The acquired information includes areas such as social security number, name, address, gender, job title, and date of birth. This data is authenticated using identity verification protocols and address validation services. Periodic audits guarantee precision and thoroughness by comparing and verifying transaction and customer records.

The Transactions Details and Customer Information tables are integrated into a secure relational database system or data warehouse through the use of primary keys, specifically the credit card number and account number. The accuracy, comprehensiveness, and currency of the dataset are ensured through real-time validation, periodic reviews, data cleaning procedures, and dynamic updates, thus maintaining data quality for rigorous analysis and informed decision-making.

Data Description:

To facilitate the analysis, we performed the necessary data integration by merging the individual transaction CSV files into a unified transactions dataset. This merging process involved combining corresponding transactional attributes from each file to create a comprehensive dataset that includes all transaction records. The details of this process are outlined in the next section.

Additionally, the transaction dataset has 4260994 observations and 10 features and the customer information dataset has 1000 observations and 9 features.

Both dataset provide crucial information for predicting total spending amounts for next month. The transaction data offers insights into spending behavior, transaction patterns as well as potential fraud indicators, while the customer data provides demographic and account-related details that could influence spending habits. By combining these datasets, the model can leverage a wide range of features to make accurate predictions.

a. Transactions Details Table

- cc_num: Credit Card Number. This field is used to identify the specific credit card involved in the transaction.
- acct_num: Account Number. This field represents the bank account associated with the transaction.
- trans_num: Transaction Number. This is a unique identifier for each transaction.
- unix_time: Unix Timestamp. This represents the date and time of the transaction in Unix time format (seconds since January 1, 1970).
- category: Transaction Category. This field specifies the type of transaction, such as "Food", "Travel", "Shopping", etc.
- amt: Amount. This is the monetary value of the transaction.
- is_fraud: Fraud Indicator. A binary field where '1' indicates the transaction is fraudulent and '0' indicates it is not.
- merchant: Merchant Name. The name of the merchant where the transaction took place.
- merch_lat: Merchant Latitude. The geographic latitude of the merchant's location.
- merch_long: Merchant Longitude. The geographic longitude of the merchant's location.

This table provides insights into transaction behaviors, patterns of spending, and potential fraud detection.

b. Customer Information Table

- ssn: Social Security Number. A unique number assigned to each individual, used for identity verification.
- cc_num: Credit Card Number. The same field as in the Transactions Details table, used to link personal details to transaction records.
- first: First Name. The customer's first name.
- last: Last Name. The customer's last name.
- gender: Gender. The gender of the customer, typically indicated as 'Male' or 'Female'.
- street: Street Address. The customer's residential street address.
- city: City. The city where the customer resides.
- state: State. The state where the customer resides.

- zip: Zip Code. The postal code for the customer's address.
- lat: Latitude. The geographic latitude of the customer's residence.
- long: Longitude. The geographic longitude of the customer's residence.
- city_pop: City Population. The population of the city where the customer resides.
- job: Job Title. The customer's occupation or job title.
- dob: Date of Birth. The birth date of the customer.
- acct_num: Account Number. The same field as in the Transactions Details table, used to link personal details to transaction records.

Key Relationships

- The cc_num field serves as a primary link between the Transactions Details and Customer Information tables, allowing for the association of transaction records with the respective customer's personal details.
- The acct_num field can also be used to link transactions to specific customer accounts, facilitating account-level analysis.

The transaction data offers insights into spending behavior, transaction patterns as well as potential fraud indicators, while the customer data provides demographic and account-related details that could influence spending habits. By combining these datasets, the model can leverage a wide range of features to make accurate predictions.

Exploratory Data Analysis (EDA):

To explore the target variable ‘amt_spend_per_month,’ the 'month' column was derived from the ‘unixtime’ column, and ‘amt_spend_per_month’ was calculated by grouping data based on ‘cc_num’ and ‘acct_num’.

Exploration of Target Variable for Business Use Case 01

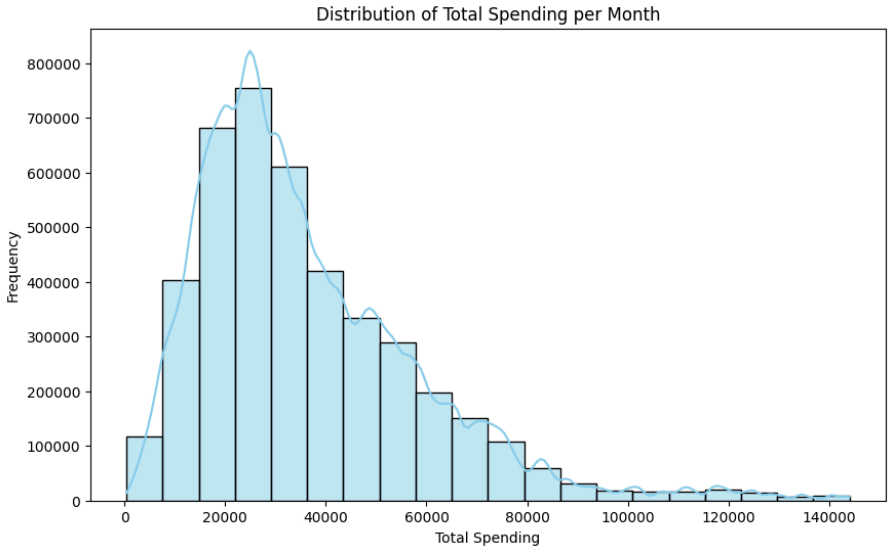


Figure 3.1 Distribution of Total Spending per Month

```
print("Statistical Summary of Total Spending per Month:")
print(merged_df['amt_spend_per_mnth'].describe())
```

```
Statistical Summary of Total Spending per Month:
count    4.260904e+06
mean     3.639086e+04
std      2.238680e+04
min      3.964600e+02
25%      2.053873e+04
50%      3.090353e+04
75%      4.798892e+04
max      1.440309e+05
Name: amt_spend_per_mnth, dtype: float64
```

Figure 3.2 Statistical Summary of Total Spending per Month

Exploration of Variables of Interest for Business Use Case 1:

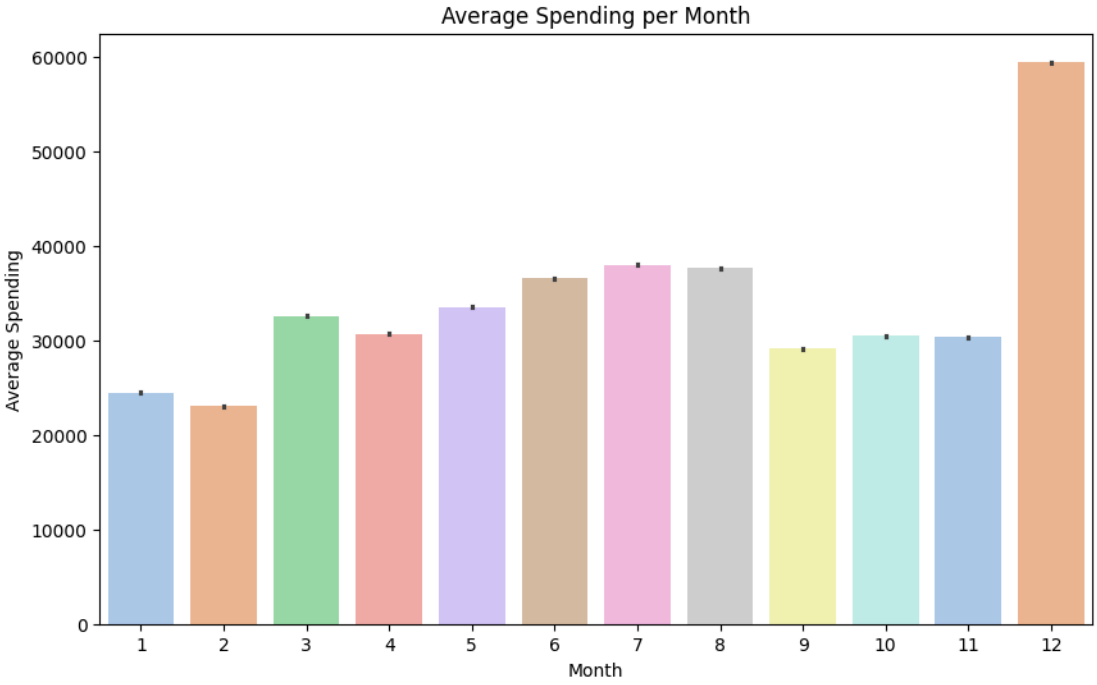


Figure 3.3 A bar chart illustrating the average spending for each month

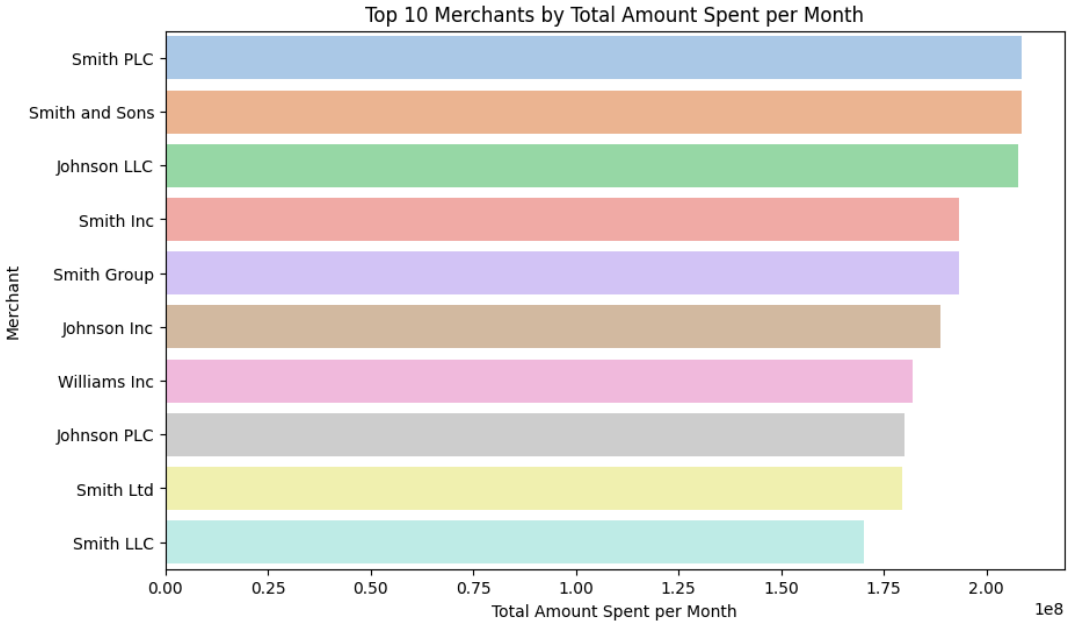


Figure 3.4 : A vertical bar chart illustrating the total amount spend per month for top 10 Merchant

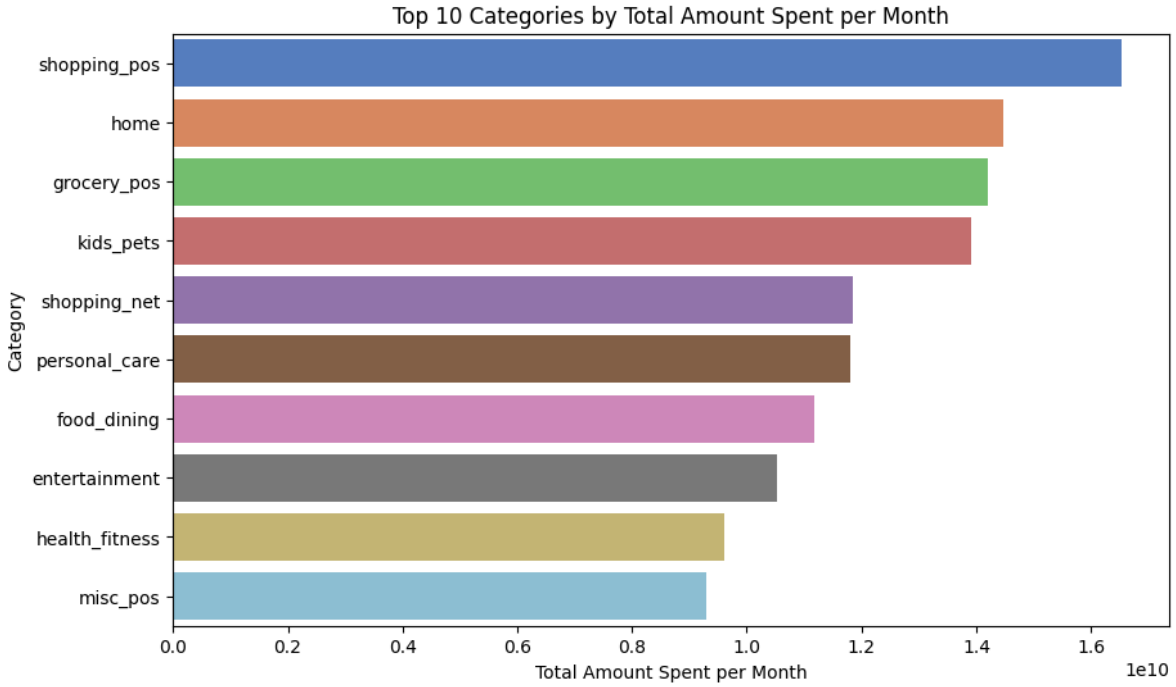


Figure 3.5 : A vertical bar chart illustrating the total amount spend per month for each top 10 category

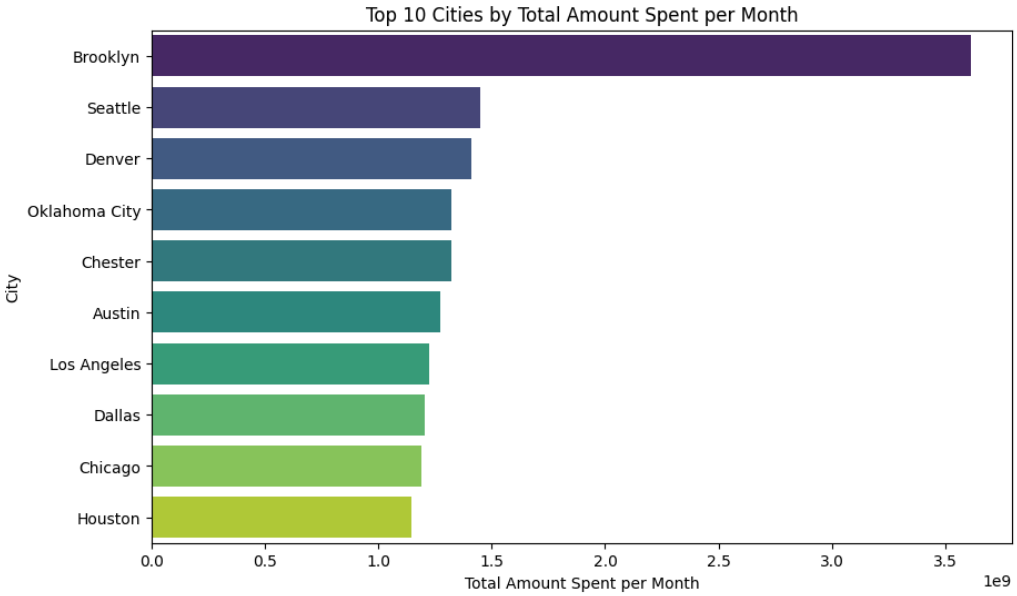


Figure 3.6: A vertical chart illustrating total amount spend per month for each top 10 city

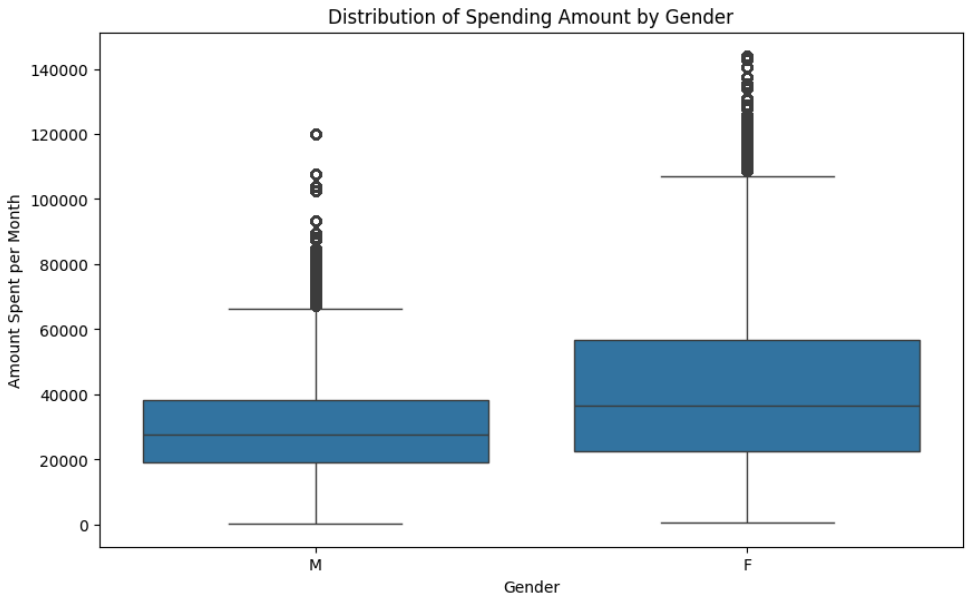


Figure 3.7: Box plot visualizing the relationship between gender and monthly spending

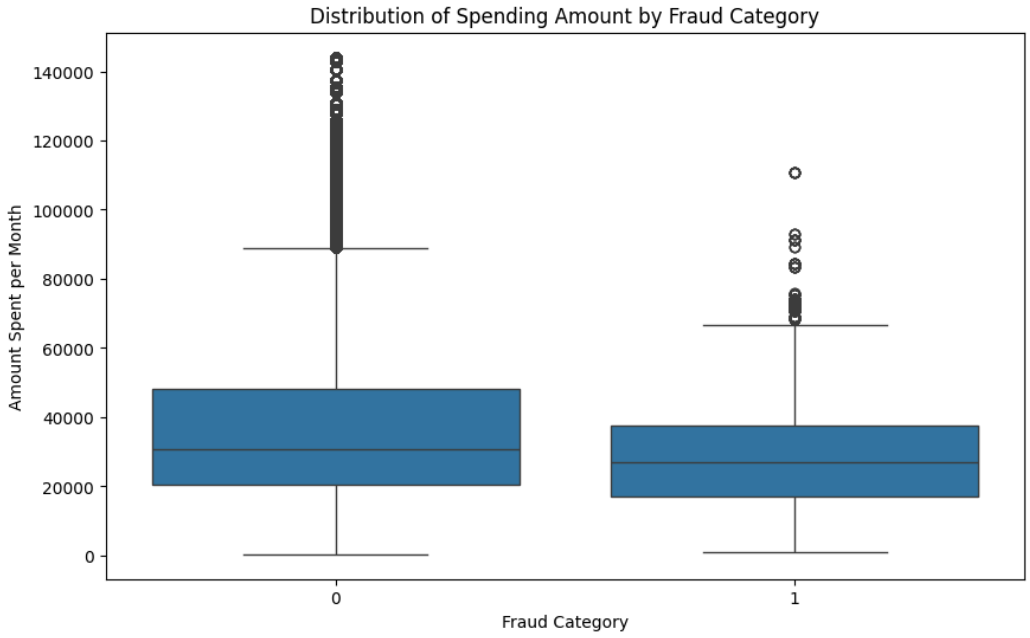


Figure 3.8 Box plot visualizing the relationship between fraud category and monthly spending

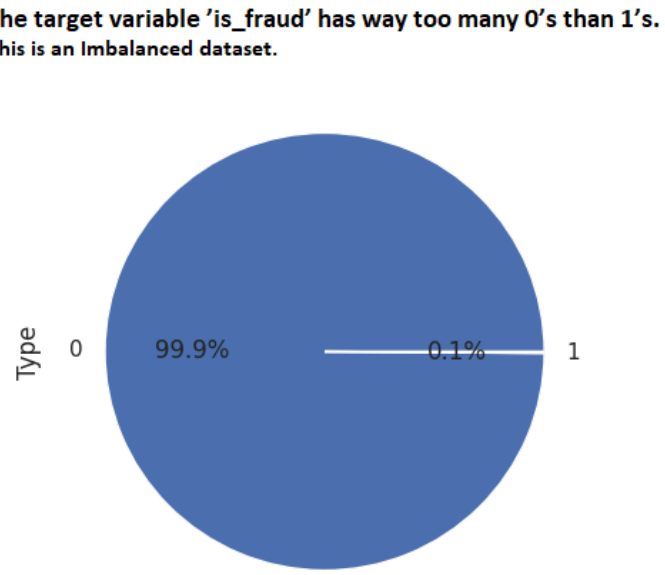


Figure 3.9 - representation of fraud in a pie chart from the dataset

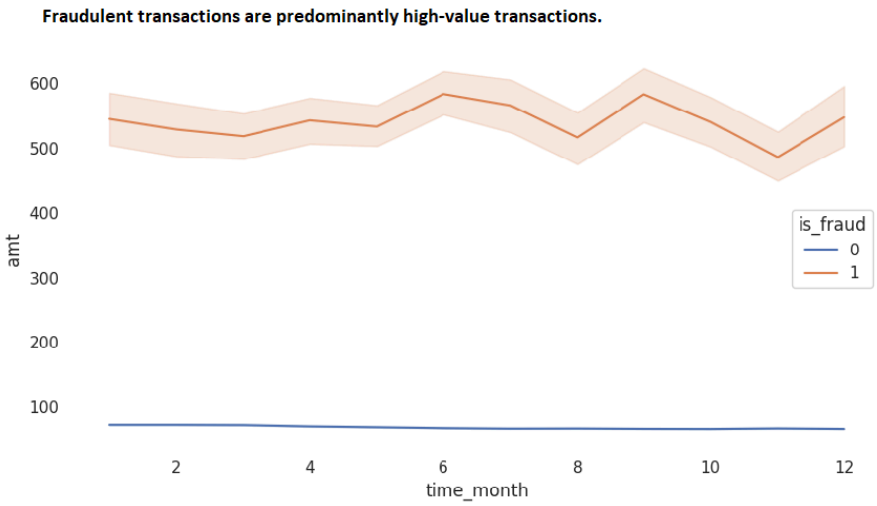


Figure 3.10 - representation of fraud in a line chart with regards to relationship between amnt and time_month from the dataset

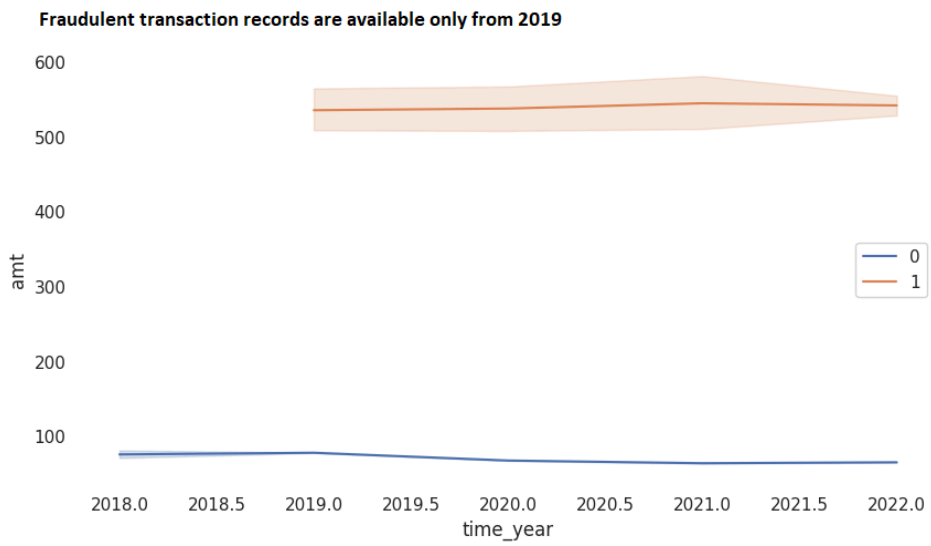


Figure 3.11 - representation of fraud in a line chart with regards to relationship between amnt and time_year variables from the dataset

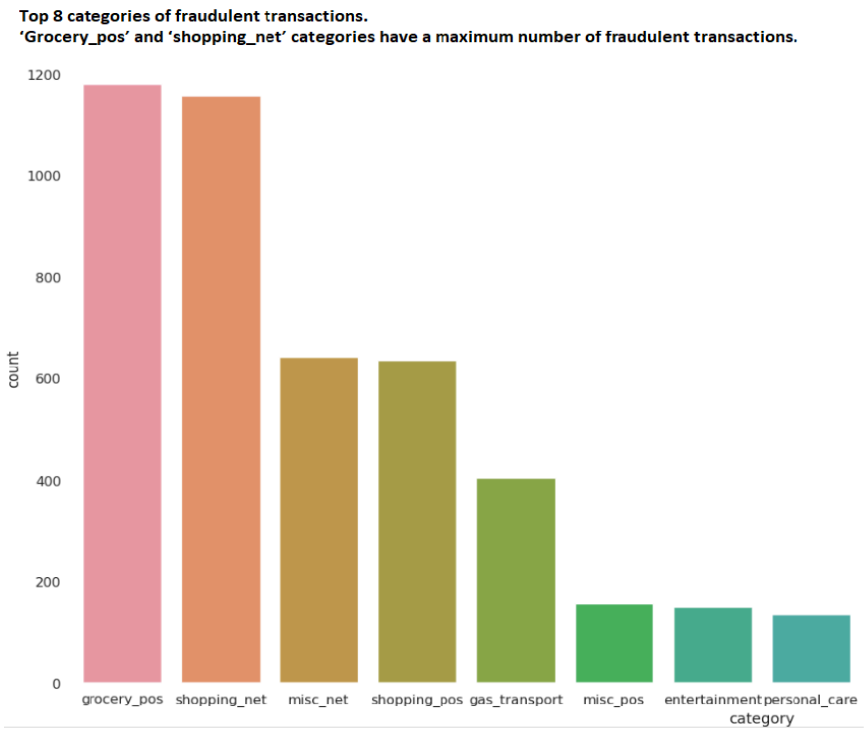


Figure 3.12 - representation of fraud in a bar chart with regards to counts for various categories of transactions from the dataset

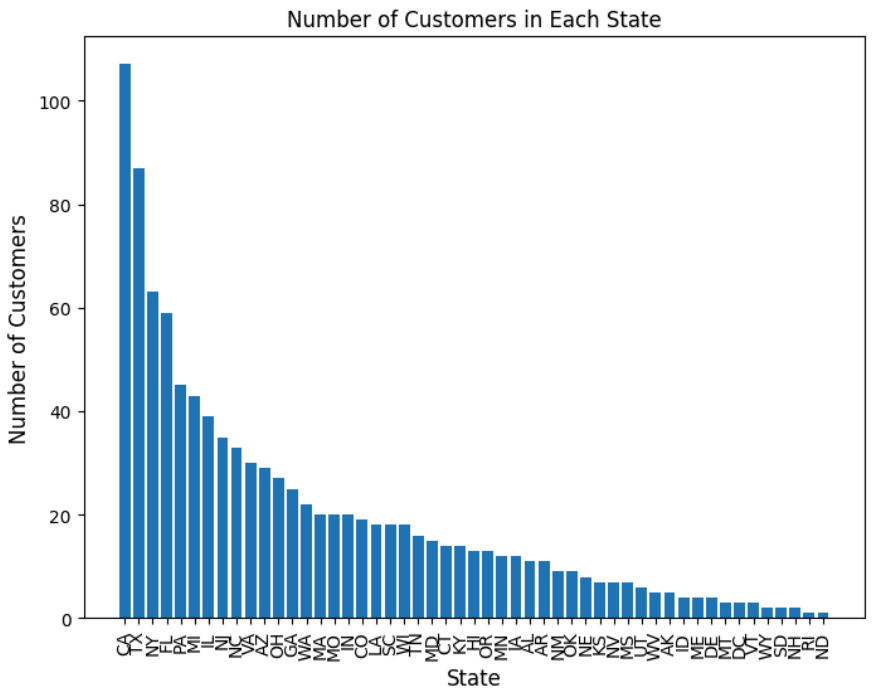


Figure 3.13 - representation of customers in different states as per the bar chart from the dataset

Variables/Features and their Importance:

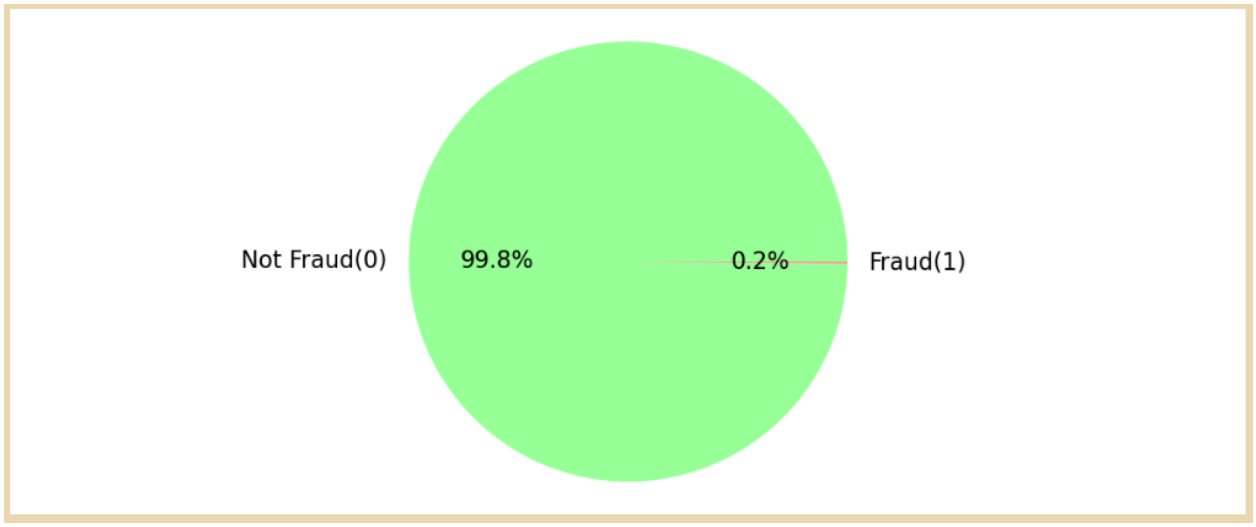


Figure 3.14 - Distribution of Target Variable(Business Case 02)

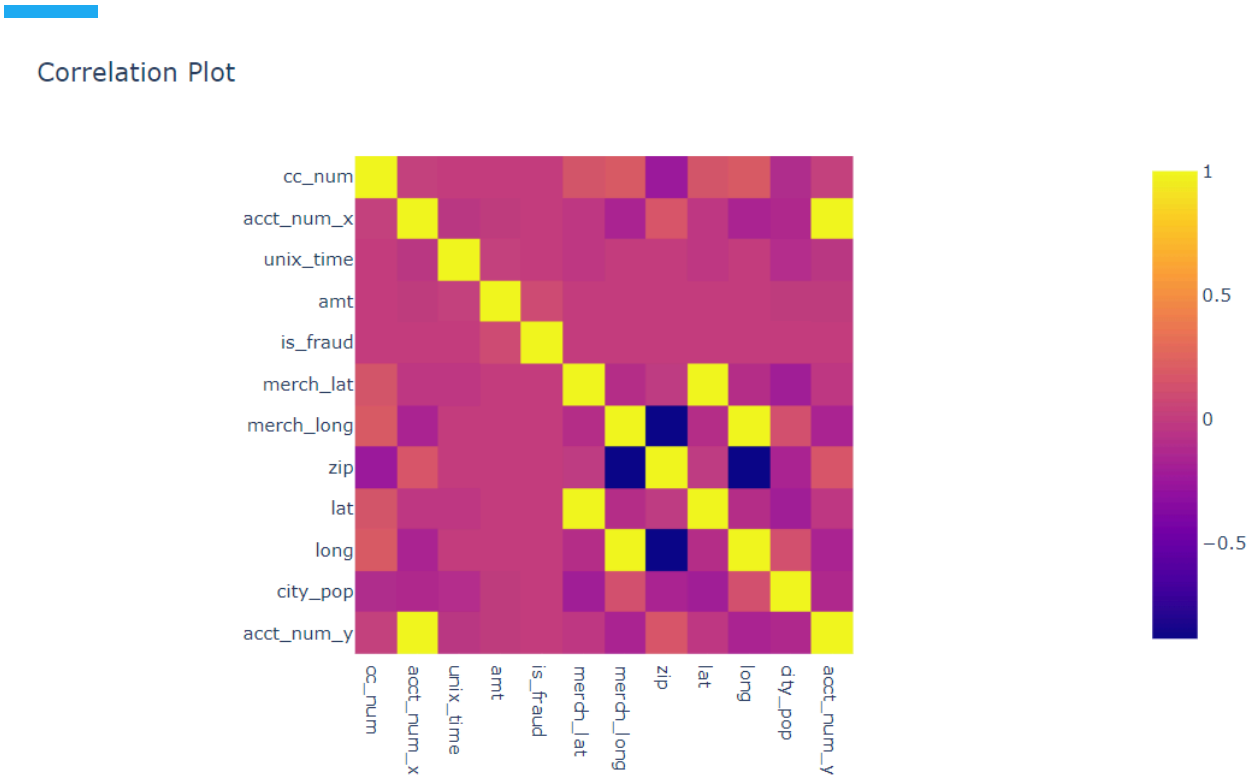


Figure 3.15 - Measure Of Dependence or Feature Correlation: (For Business Case 02)

Correlation heatmap

Colormap: Most negative correlations (dark-blue) to most positive correlation (dark red)

	cc_num	acct_num_x	unix_time	amt	is_fraud	merch_lat	merch_long	zip	lat	long
cc_num	1.000000	0.022088	0.005663	-0.000241	0.009796	0.145447	0.186959	-0.245161	0.145955	0.186971
acct_num_x	0.022088	1.000000	-0.051577	-0.010214	-0.005604	-0.043152	-0.160391	0.161042	-0.043484	-0.160533
unix_time	0.005663	-0.051577	1.000000	0.017001	0.004481	-0.041818	0.001352	-0.001117	-0.042045	0.001363
amt	-0.000241	-0.010214	0.017001	1.000000	0.100589	-0.004714	0.000394	-0.001459	-0.004467	0.000442
is_fraud	0.009796	-0.005604	0.004481	0.100589	1.000000	-0.000475	0.003587	-0.001836	-0.000373	0.003646
merch_lat	0.145447	-0.043152	-0.041818	-0.004714	-0.000475	1.000000	-0.085027	-0.014844	0.995882	-0.085195
merch_long	0.186959	-0.160391	0.001352	0.000394	0.003587	-0.085027	1.000000	-0.886709	-0.085500	0.999341
zip	-0.245161	0.161042	-0.001117	-0.001459	-0.001836	-0.014844	-0.886709	1.000000	-0.014702	-0.887194
lat	0.145955	-0.043484	-0.042045	-0.004467	-0.000373	0.995882	-0.085500	-0.014702	1.000000	-0.085673
long	0.186971	-0.160533	0.001363	0.000442	0.003646	-0.085195	0.999341	-0.887194	-0.085673	1.000000
city_pop	-0.108829	-0.139880	-0.105473	-0.009276	0.005865	-0.207405	0.132450	-0.166511	-0.208438	0.132588
acct_num_y	0.022088	1.000000	-0.051577	-0.010214	-0.005604	-0.043152	-0.160391	0.161042	-0.043484	-0.160533

Figure 3.16 - Correlation between variables

merch_long	zip	lat	long	city_pop	acct_num_y
0.186959	-0.245161	0.145955	0.186971	-0.108829	0.022088
-0.160391	0.161042	-0.043484	-0.160533	-0.139880	1.000000
0.001352	-0.001117	-0.042045	0.001363	-0.105473	-0.051577
0.000394	-0.001459	-0.004467	0.000442	-0.009276	-0.010214
0.003587	-0.001836	-0.000373	0.003646	0.005865	-0.005604
-0.085027	-0.014844	0.995882	-0.085195	-0.207405	-0.043152
1.000000	-0.886709	-0.085500	0.999341	0.132450	-0.160391
-0.886709	1.000000	-0.014702	-0.887194	-0.166511	0.161042
-0.085500	-0.014702	1.000000	-0.085673	-0.208438	-0.043484
0.999341	-0.887194	-0.085673	1.000000	0.132588	-0.160533
0.132450	-0.166511	-0.208438	0.132588	1.000000	-0.139880
-0.160391	0.161042	-0.043484	-0.160533	-0.139880	1.000000

Figure 3.17 - Correlation between variables

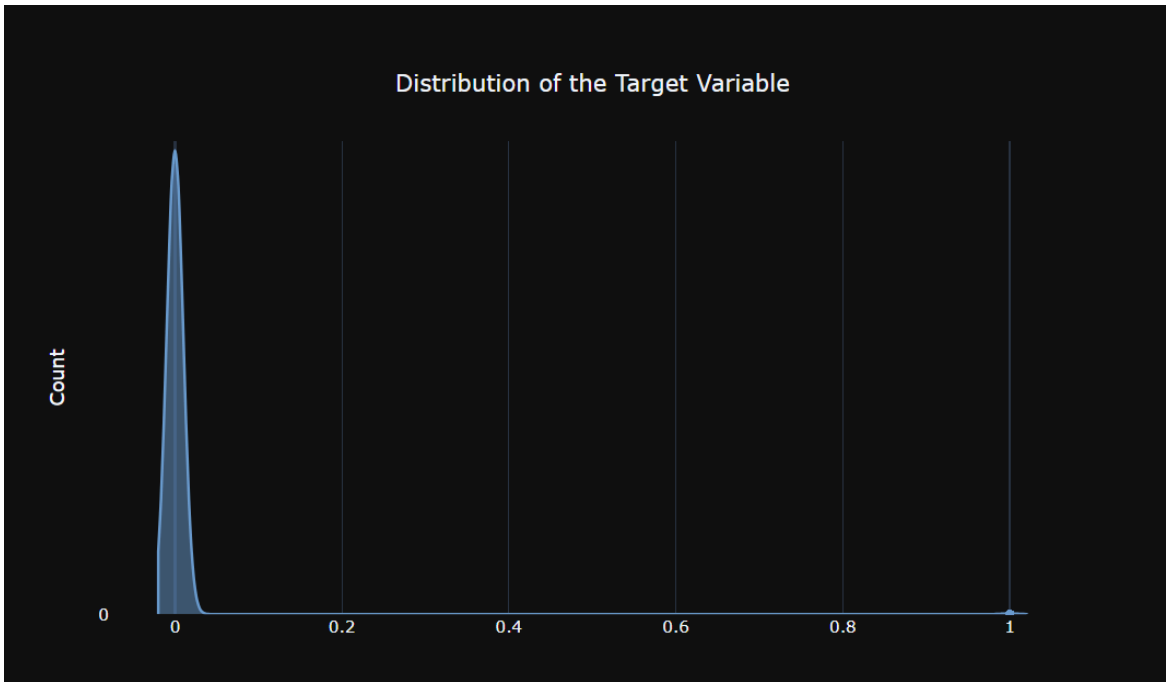


Figure 3.18 - distribution of target variable ***(For Business Case 02)***

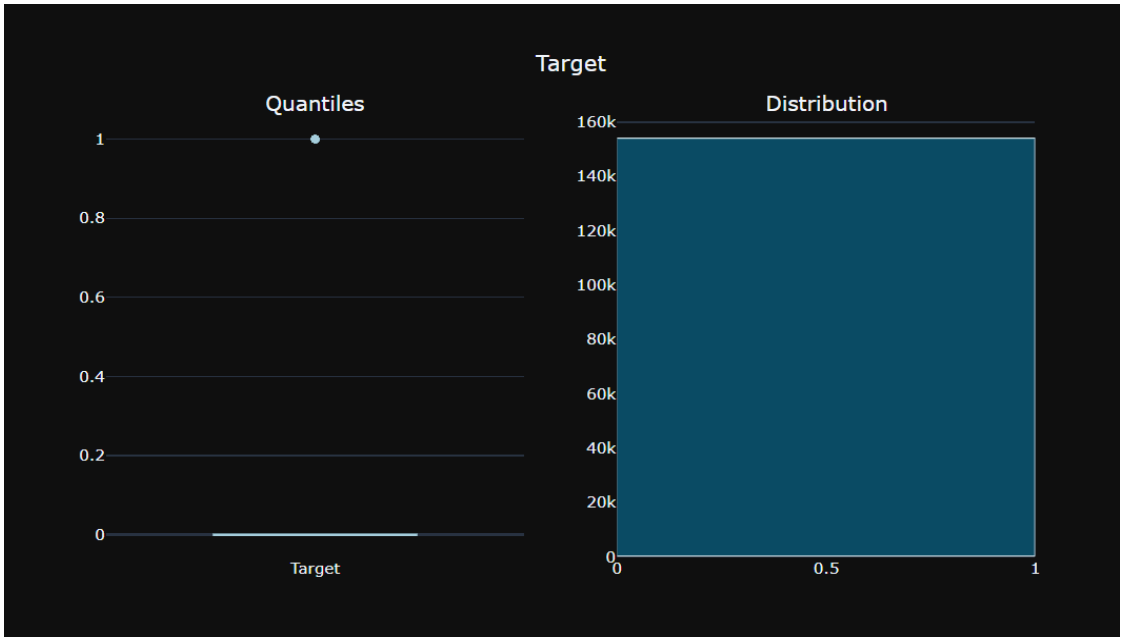


Figure 3.19 - distribution of target variable ***(For Business Case 02)***

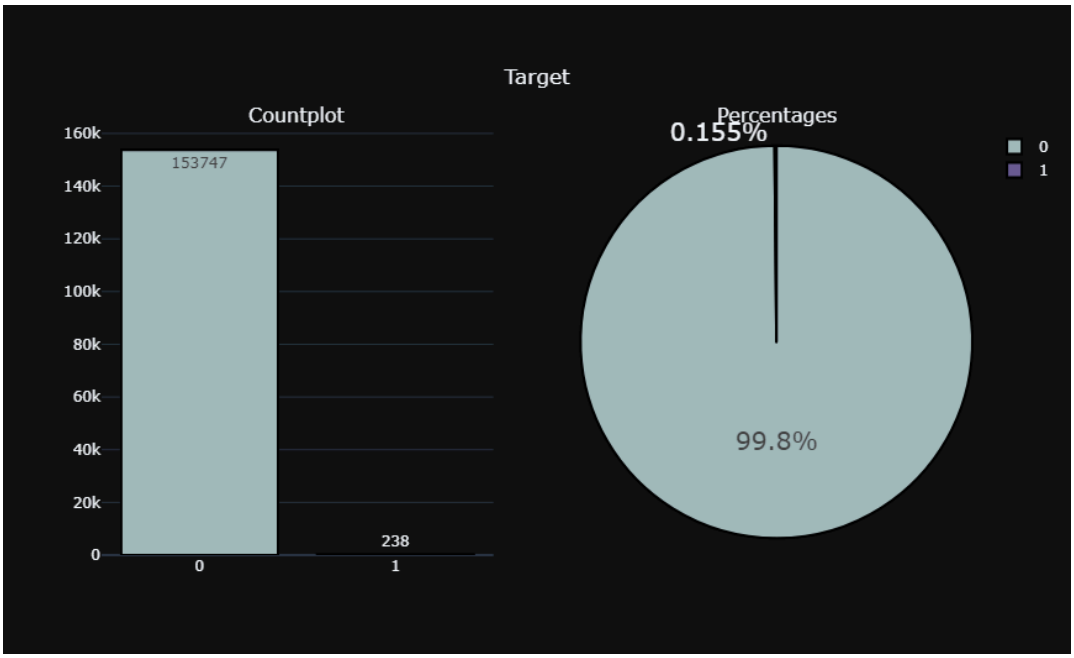


Figure 3.20 - distribution of target variable ***(For Business Case 02)***

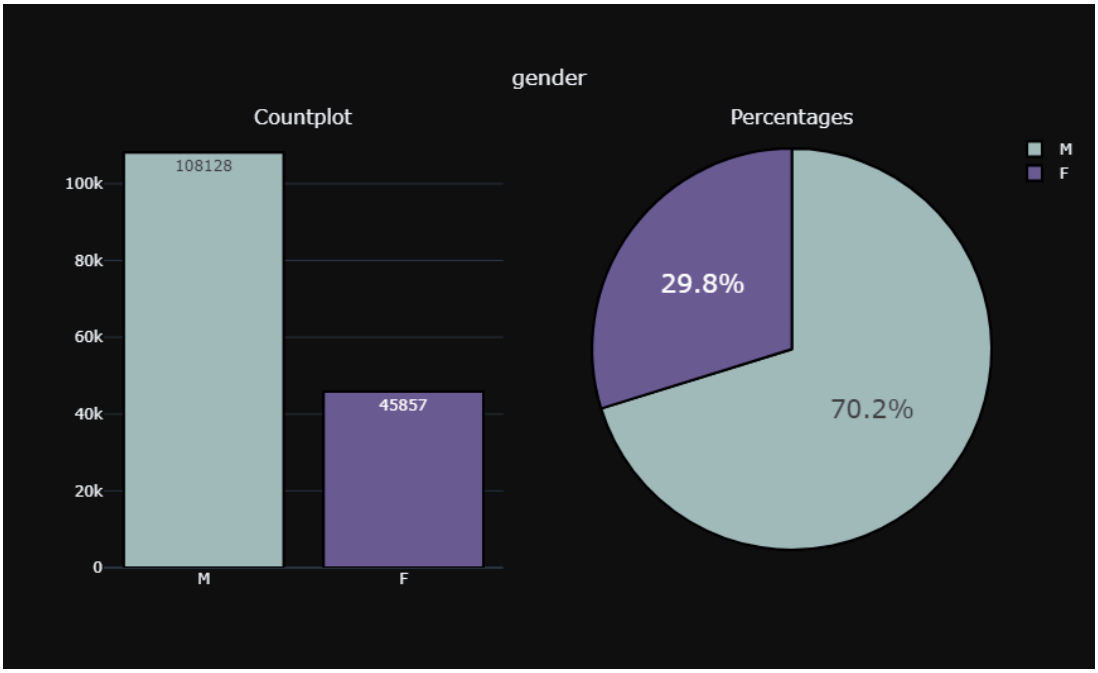


Figure 3.21 - distribution of gender values from the dataset ***(For Business Case 02)***

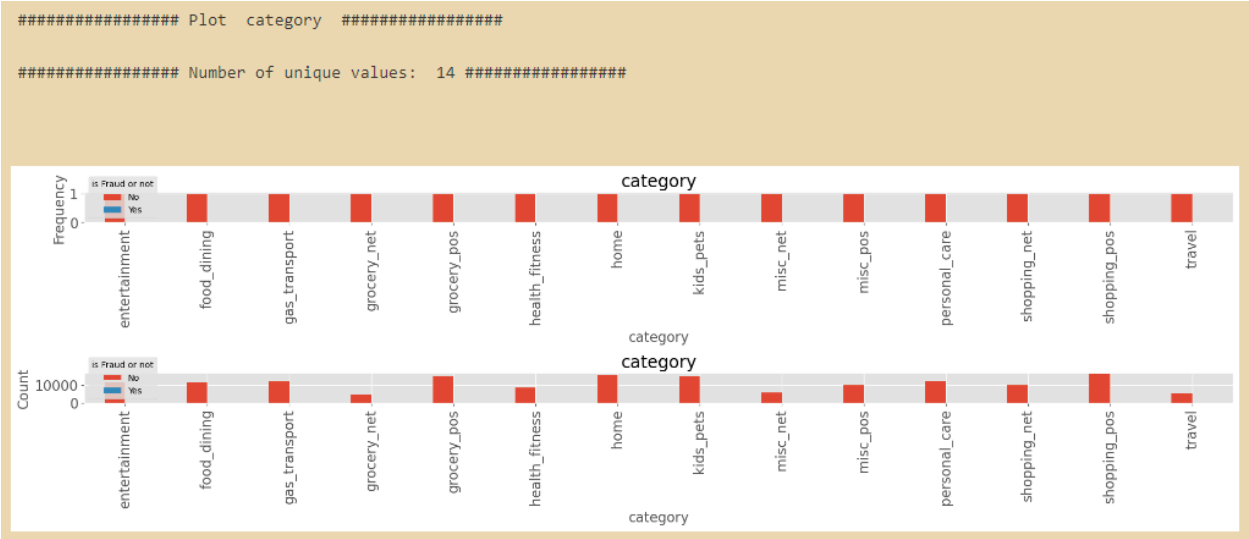


Figure 3.22 - Target vs category (For Business Case 02)

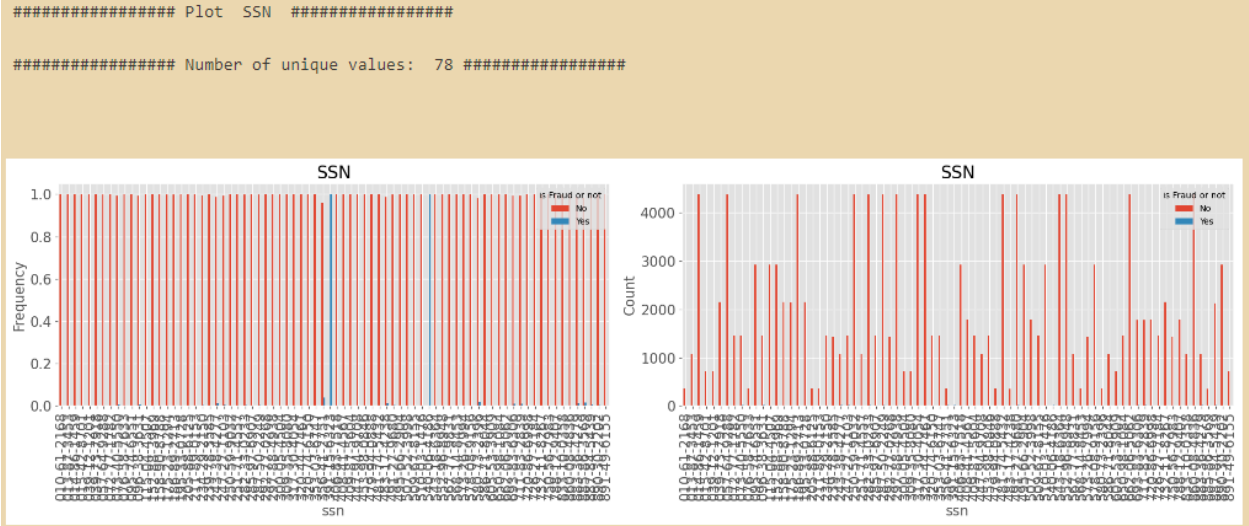


Figure 3.23 - Target vs category (For Business Case 02)

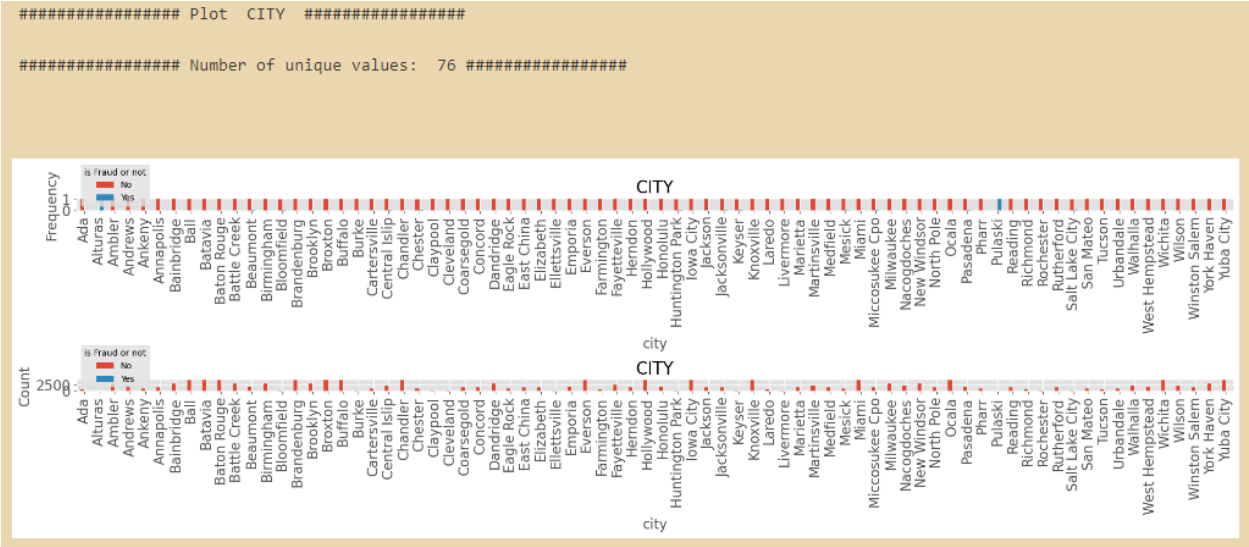


Figure 3.24 - Target vs City (For Business Case 02)

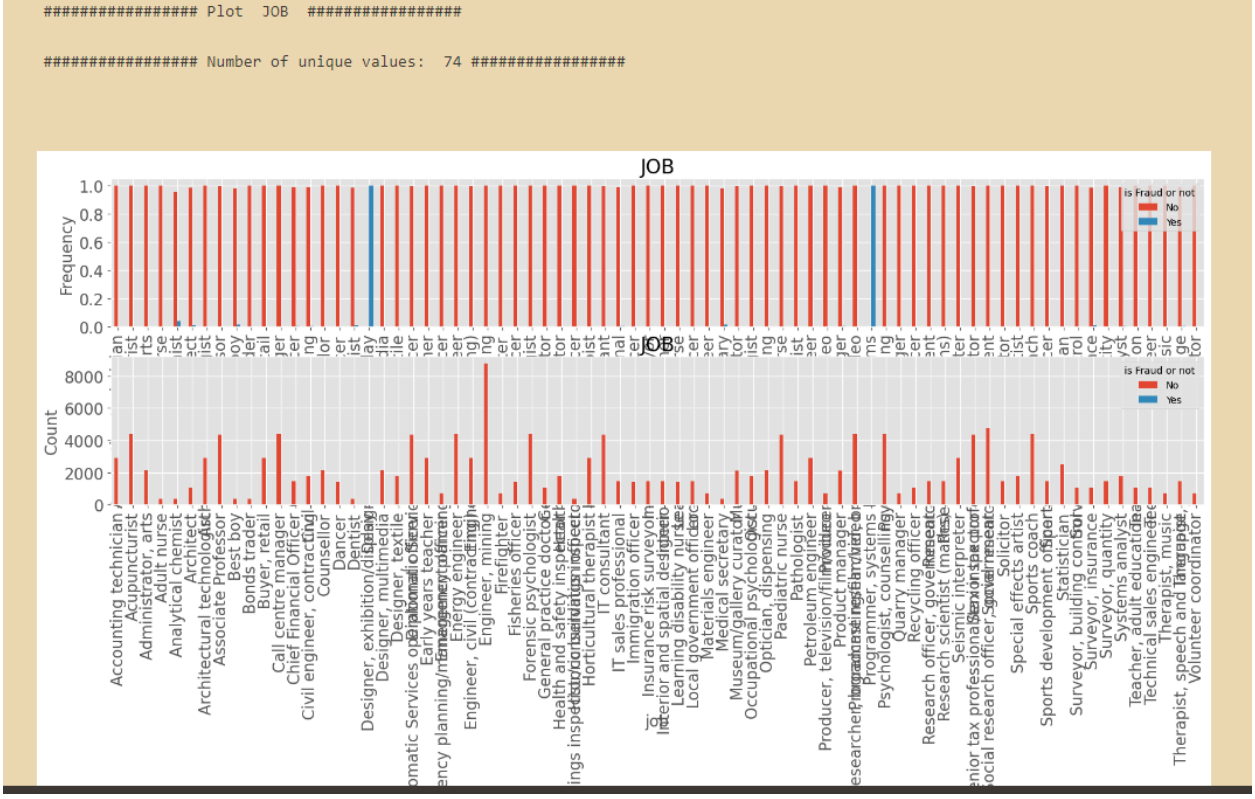


Figure 3.25 - Target vs Job (For Business Case 02)

Counter[plots for business Use Case 02

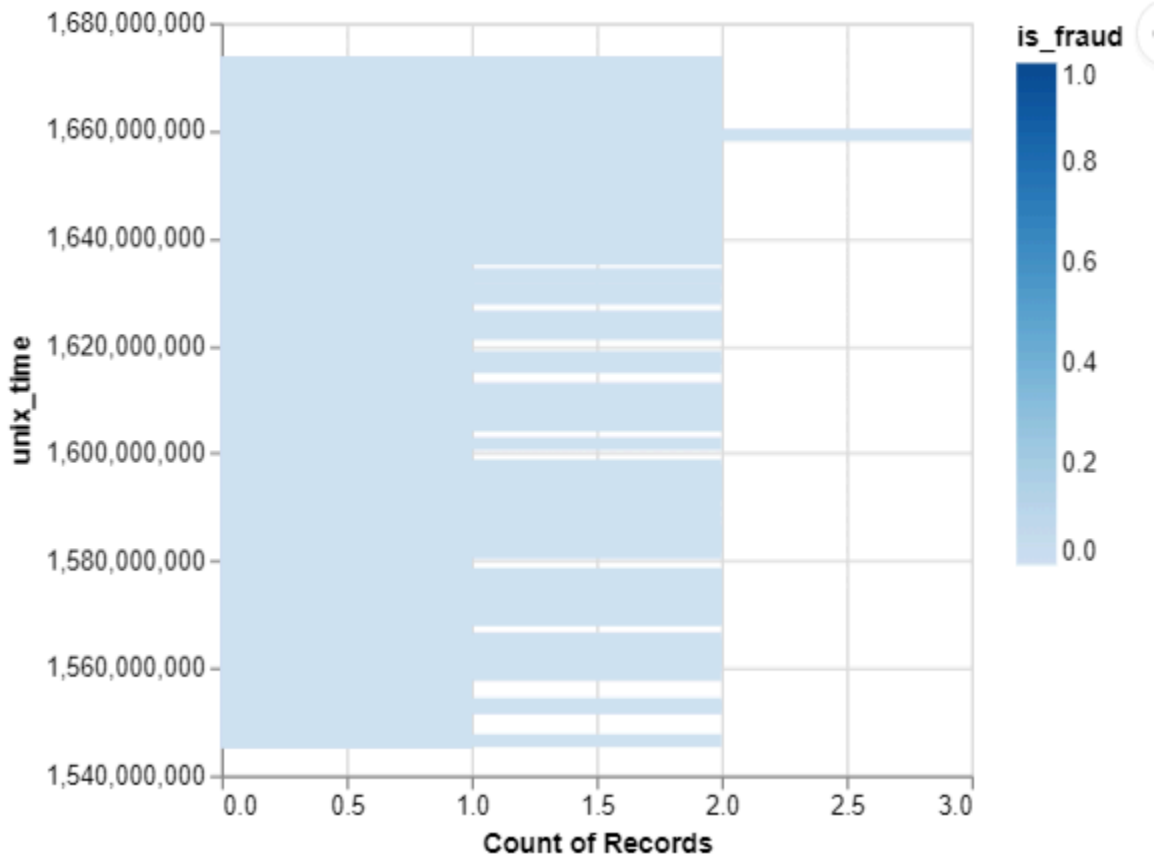
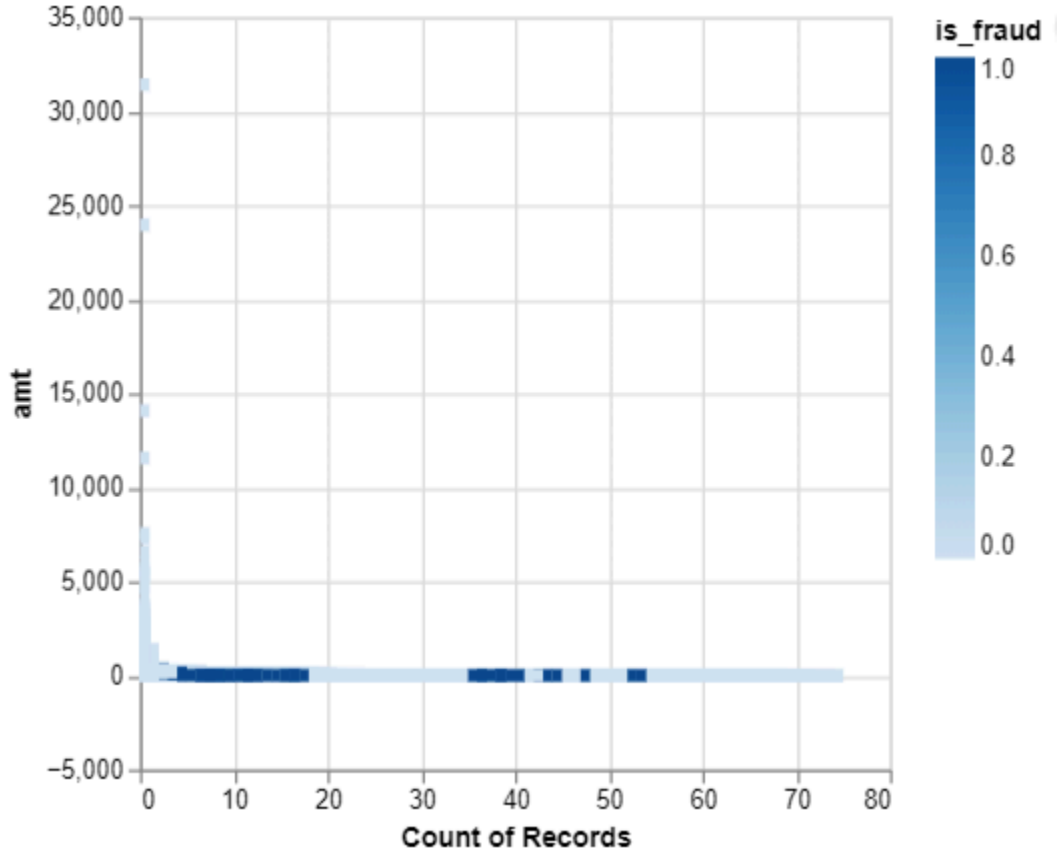
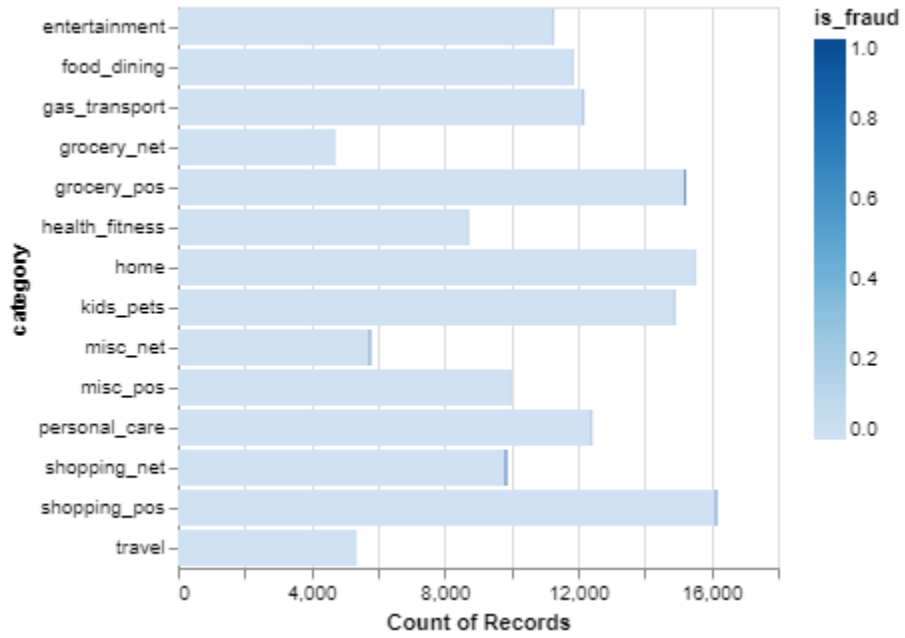
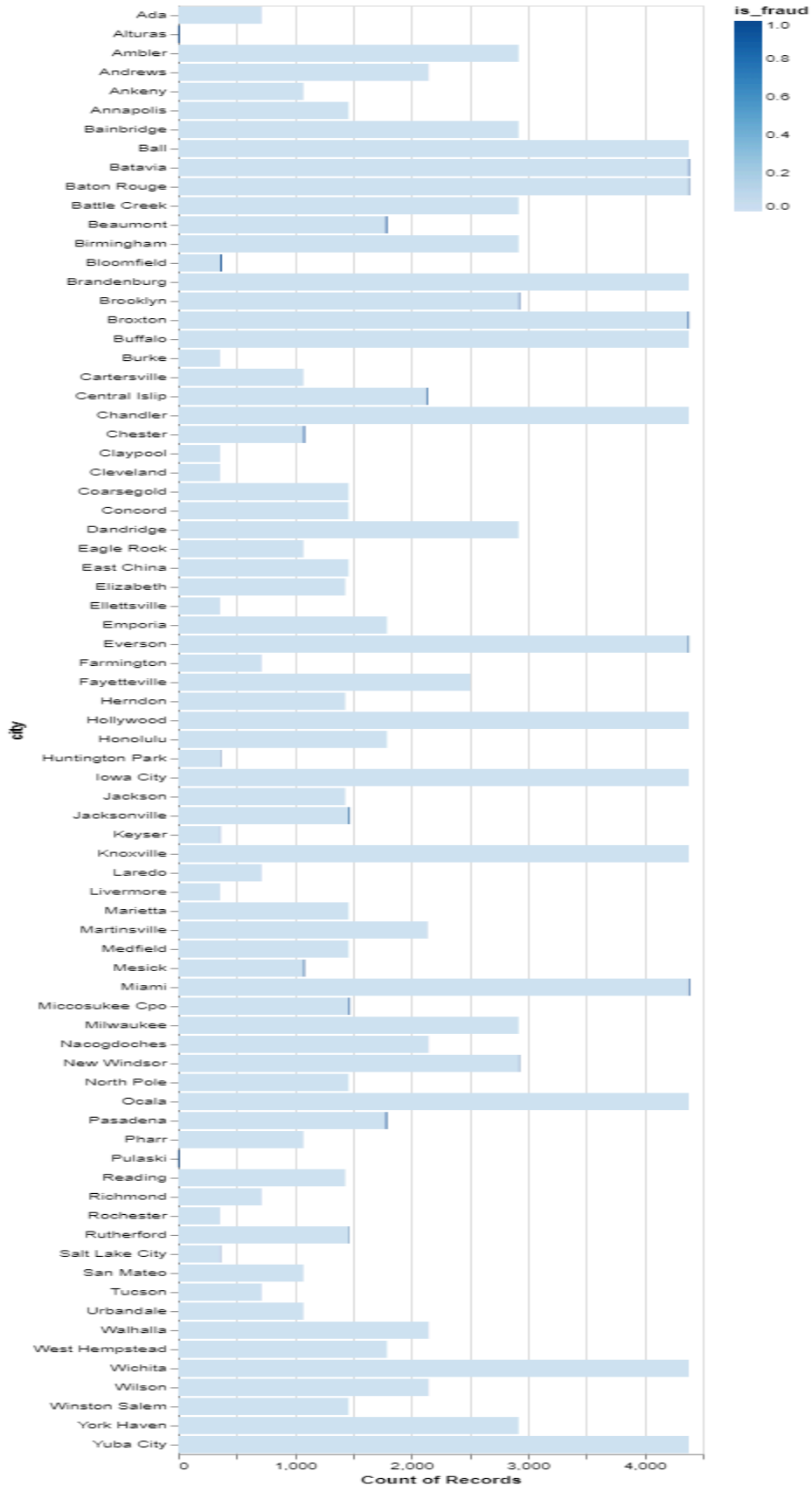
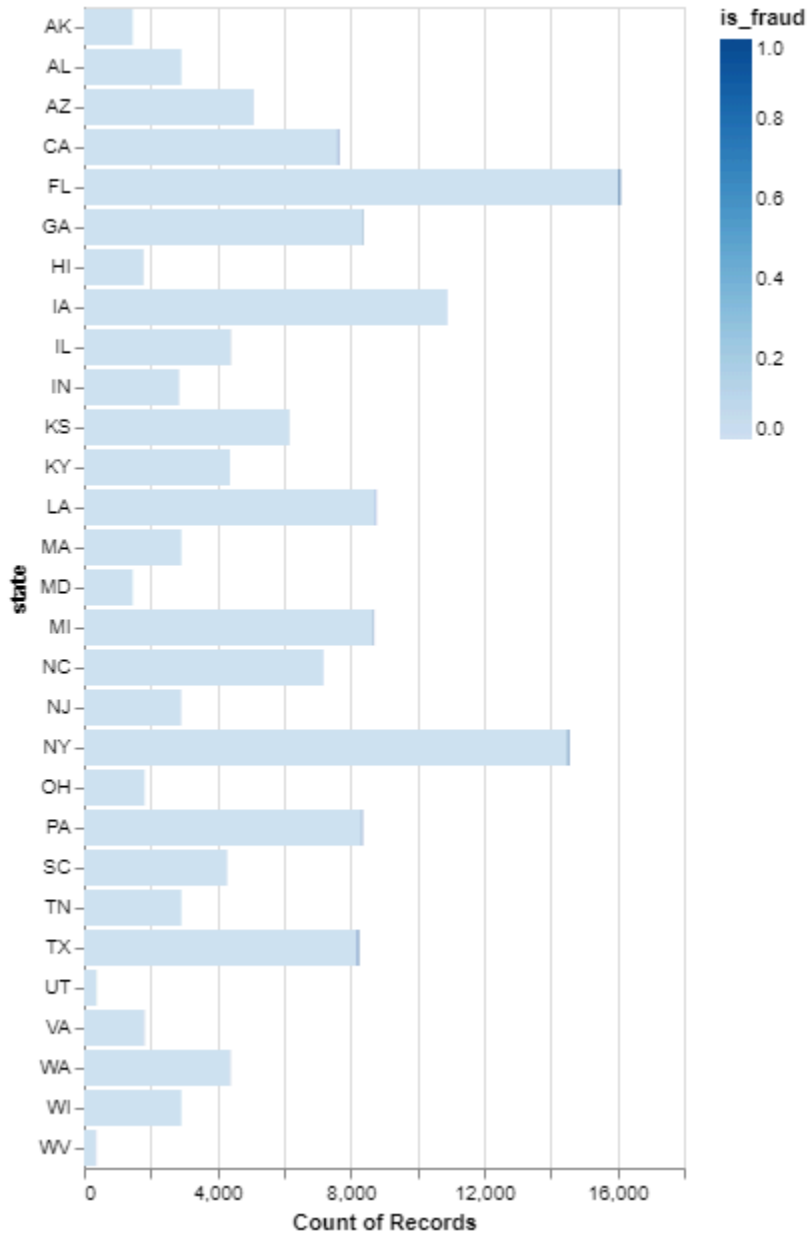


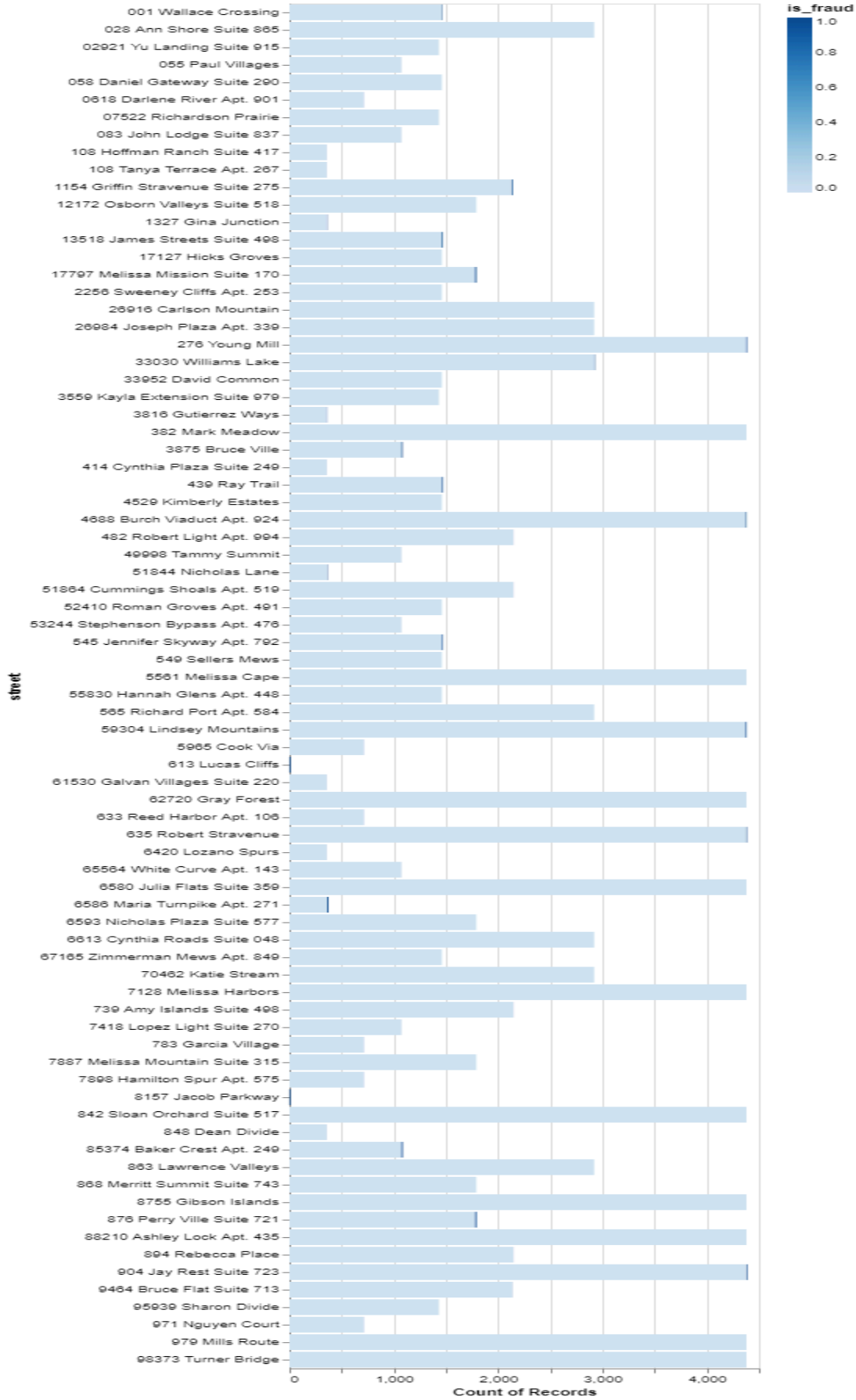
Figure - Target (For Business Case 02)

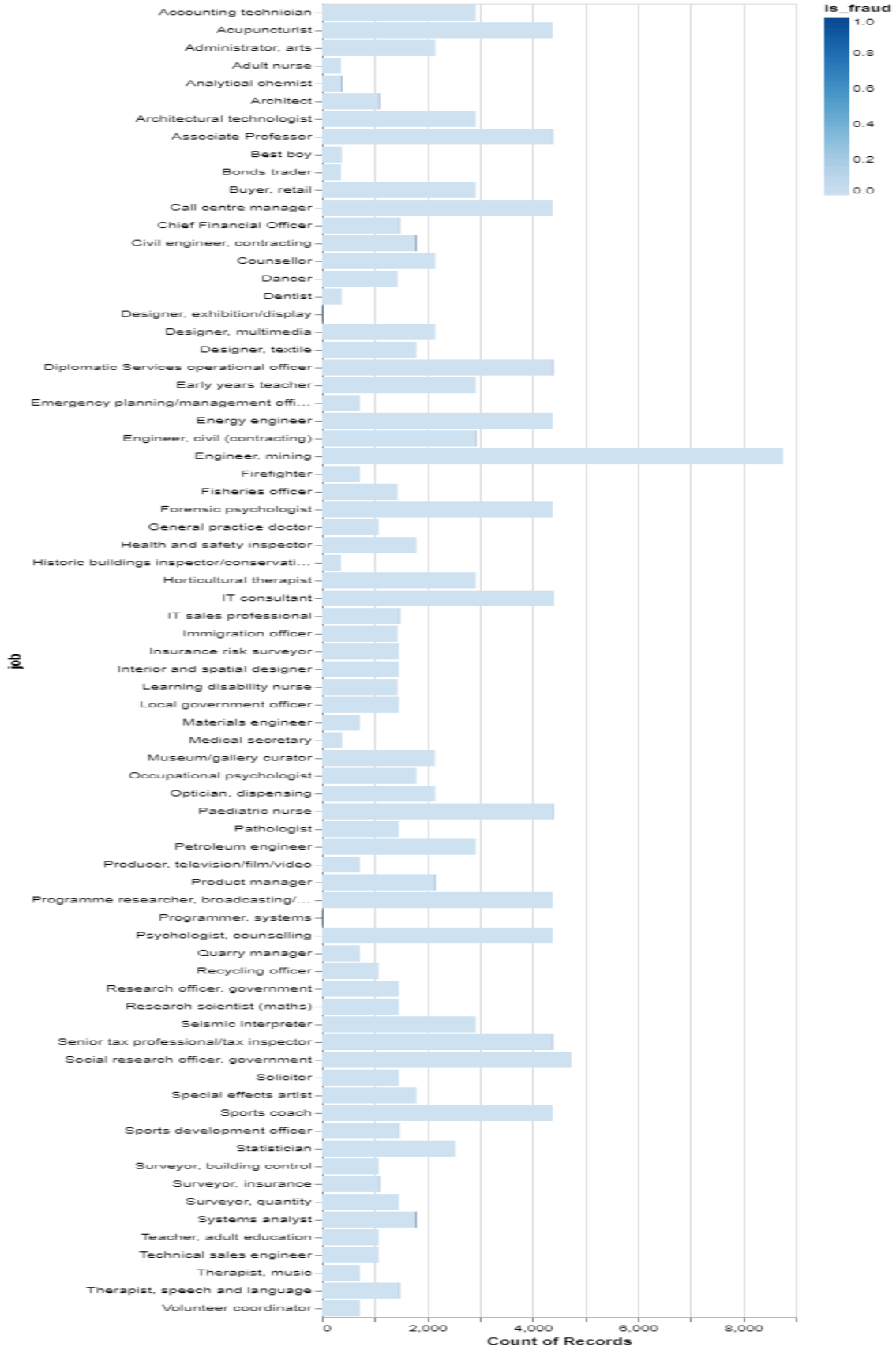

35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
−5,000
amt
0
10
20
30
40
50
60
70
80
Count of Records
is_fraud
1.0
0.8
0.6
0.4
0.2
0.0

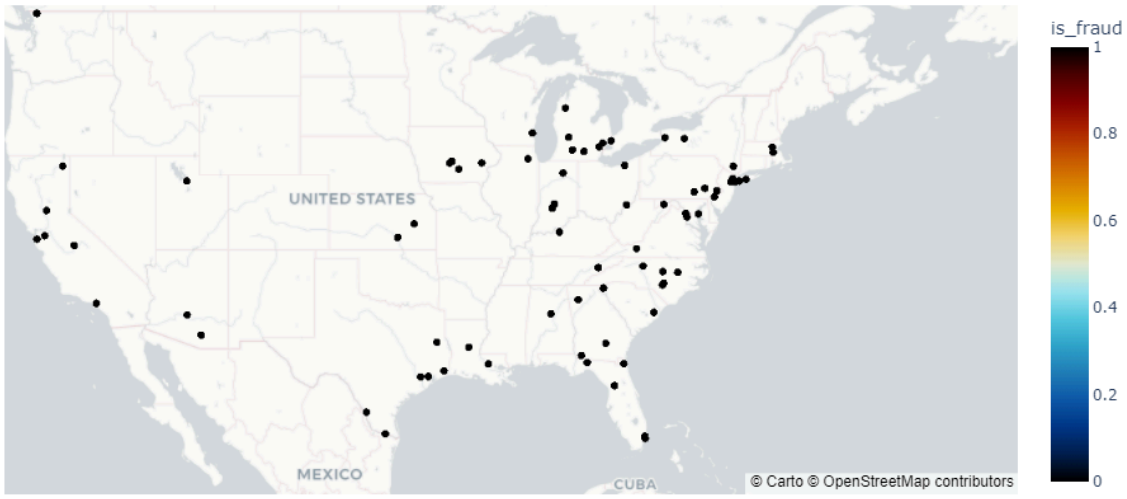

is_fraud
1.0
0.8
0.6
0.4
0.2
0.0
category
entertainment
food_dining
gas_transport
grocery_net
grocery_pos
health_fitness
home
kids_pets
misc_net
misc_pos
personal_care
shopping_net
shopping_pos
travel
0
4,000
8,000
12,000
16,000
Count of Records


is_fraud
1.0
0.8
0.6
0.4
0.2
0.0
city
Ada
Alturas
Ambler
Andrews
Ankeny
Annapolis
Bainbridge
Ball
Batavia
Baton Rouge
Battle Creek
Beaumont
Birmingham
Bloomfield
Brandenburg
Brooklyn
Broxton
Buffalo
Burke
Cartersville
Central Islip
Chandler
Chester
Claypool
Cleveland
Coarsegold
Concord
Dandridge
Eagle Rock
East China
Elizabeth
Ellettsville
Emporia
Everson
Farmington
Fayetteville
Herndon
Hollywood
Honolulu
Huntington Park
Iowa City
Jackson
Jacksonville
Keyser
Knoxville
Laredo
Livermore
Marietta
Martinsville
Medfield
Mesick
Miami
Miccosukee Cpo
Milwaukee
Nacogdoches
New Windsor
North Pole
Ocala
Pasadena
Pharr
Pulaski
Reading
Richmond
Rochester
Rutherford
Salt Lake City
San Mateo
Tucson
Urbandale
Walhalla
West Hempstead
Wichita
Wilson
Winston Salem
York Haven
Yuba City
0
1,000
2,000
3,000
4,000
Count of Records


is_fraud
1.0
0.8
0.6
0.4
0.2
0.0
state
AK
AL
AZ
CA
FL
GA
HI
IA
IL
IN
KS
KY
LA
MA
MD
MI
NC
NJ
NY
OH
PA
SC
TN
TX
UT
VA
WA
WI
WV
0
4,000
8,000
12,000
16,000
Count of Records


is_fraud
1.0
0.8
0.6
0.4
0.2
0.0
street
001 Wallace Crossing
028 Ann Shore Suite 865
02921 Yu Landing Suite 915
055 Paul Villages
058 Daniel Gateway Suite 290
0618 Darlene River Apt. 901
07522 Richardson Prairie
083 John Lodge Suite 837
108 Hoffman Ranch Suite 417
108 Tanya Terrace Apt. 267
1154 Griffin Stravenue Suite 275
12172 Osborn Valleys Suite 518
1327 Gina Junction
13518 James Streets Suite 498
17127 Hicks Groves
17797 Melissa Mission Suite 170
2256 Sweeney Cliffs Apt. 253
26916 Carlson Mountain
26984 Joseph Plaza Apt. 339
276 Young Mill
33030 Williams Lake
33952 David Common
3559 Kayla Extension Suite 979
3816 Gutierrez Ways
382 Mark Meadow
3875 Bruce Ville
414 Cynthia Plaza Suite 249
439 Ray Trail
4529 Kimberly Estates
4688 Burch Viaduct Apt. 924
482 Robert Light Apt. 994
49998 Tammy Summit
51844 Nicholas Lane
51864 Cummings Shoals Apt. 519
52410 Roman Groves Apt. 491
53244 Stephenson Bypass Apt. 476
545 Jennifer Skyway Apt. 792
549 Sellers Mews
5561 Melissa Cape
55830 Hannah Glens Apt. 448
565 Richard Port Apt. 584
59304 Lindsey Mountains
5965 Cook Via
613 Lucas Cliffs
61530 Galvan Villages Suite 220
62720 Gray Forest
633 Reed Harbor Apt. 106
635 Robert Stravenue
6420 Lozano Spurs
65564 White Curve Apt. 143
6580 Julia Flats Suite 359
6586 Maria Turnpike Apt. 271
6593 Nicholas Plaza Suite 577
6613 Cynthia Roads Suite 048
67165 Zimmerman Mews Apt. 849
70462 Katie Stream
7128 Melissa Harbors
739 Amy Islands Suite 498
7418 Lopez Light Suite 270
783 Garcia Village
7887 Melissa Mountain Suite 315
7898 Hamilton Spur Apt. 575
8157 Jacob Parkway
842 Sloan Orchard Suite 517
848 Dean Divide
85374 Baker Crest Apt. 249
863 Lawrence Valleys
868 Merritt Summit Suite 743
8755 Gibson Islands
876 Perry Ville Suite 721
88210 Ashley Lock Apt. 435
894 Rebecca Place
904 Jay Rest Suite 723
9464 Bruce Flat Suite 713
95939 Sharon Divide
971 Nguyen Court
979 Mills Route
98373 Turner Bridge
0
1,000
2,000
3,000
4,000
Count of Records


is_fraud
1.0
0.8
0.6
0.4
0.2
0.0
job
Accounting technician
Acupuncturist
Administrator, arts
Adult nurse
Analytical chemist
Architect
Architectural technologist
Associate Professor
Best boy
Bonds trader
Buyer, retail
Call centre manager
Chief Financial Officer
Civil engineer, contracting
Counsellor
Dancer
Dentist
Designer, exhibition/display
Designer, multimedia
Designer, textile
Diplomatic Services operational officer
Early years teacher
Emergency planning/management offi...
Energy engineer
Engineer, civil (contracting)
Engineer, mining
Firefighter
Fisheries officer
Forensic psychologist
General practice doctor
Health and safety inspector
Historic buildings inspector/conservati...
Horticultural therapist
IT consultant
IT sales professional
Immigration officer
Insurance risk surveyor
Interior and spatial designer
Learning disability nurse
Local government officer
Materials engineer
Medical secretary
Museum/gallery curator
Occupational psychologist
Optician, dispensing
Paediatric nurse
Pathologist
Petroleum engineer
Producer, television/film/video
Product manager
Programme researcher, broadcasting/...
Programmer, systems
Psychologist, counselling
Quarry manager
Recycling officer
Research officer, government
Research scientist (maths)
Seismic interpreter
Senior tax professional/tax inspector
Social research officer, government
Solicitor
Special effects artist
Sports coach
Sports development officer
Statistician
Surveyor, building control
Surveyor, insurance
Surveyor, quantity
Systems analyst
Teacher, adult education
Technical sales engineer
Therapist, music
Therapist, speech and language
Volunteer coordinator
0
2,000
4,000
6,000
8,000
Count of Records



Understanding the correlation of the features characterizing the customer's spending behavior. The features include zip code, latitude, longitude, city population, amount, merchant latitude, merchant longitude, and whether the transaction was a fraud or not. (For Business Case 03)

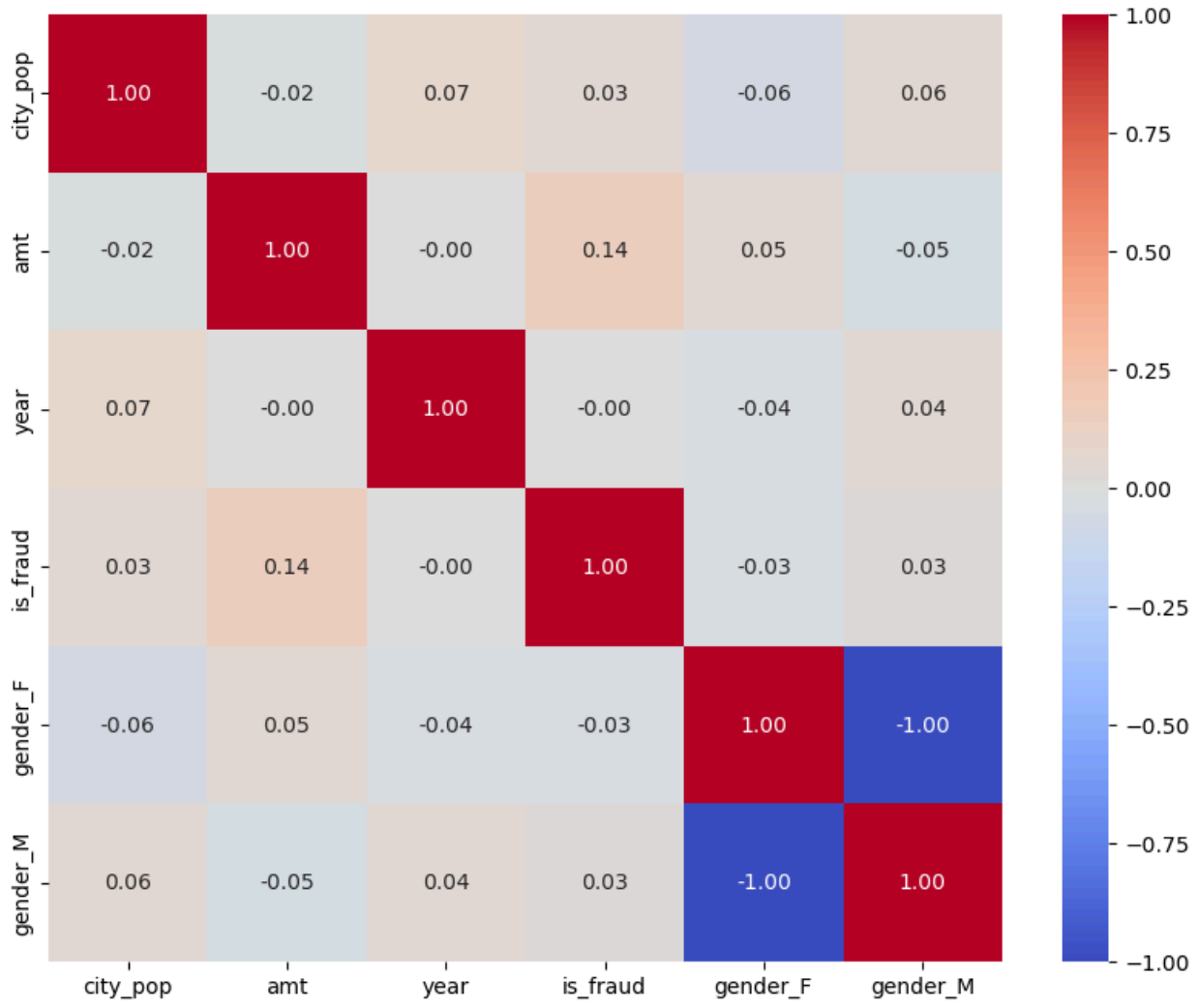


Figure 3.26 -Correlation Matrix

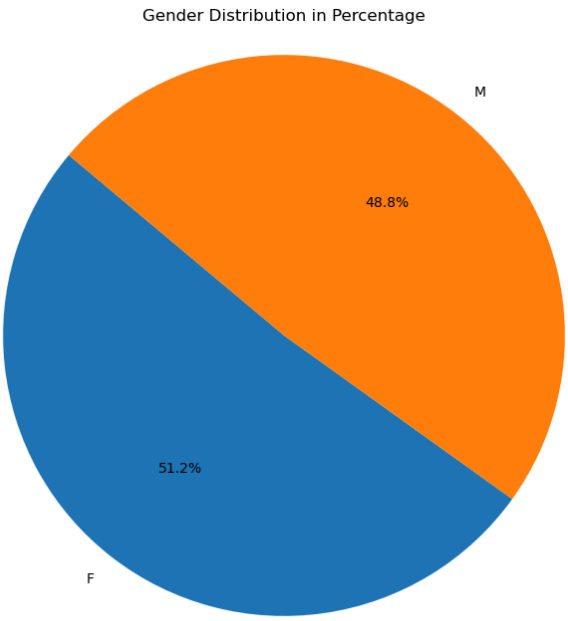


Figure 3.27 -Understanding the gender of the customer base (For Business Case 03)

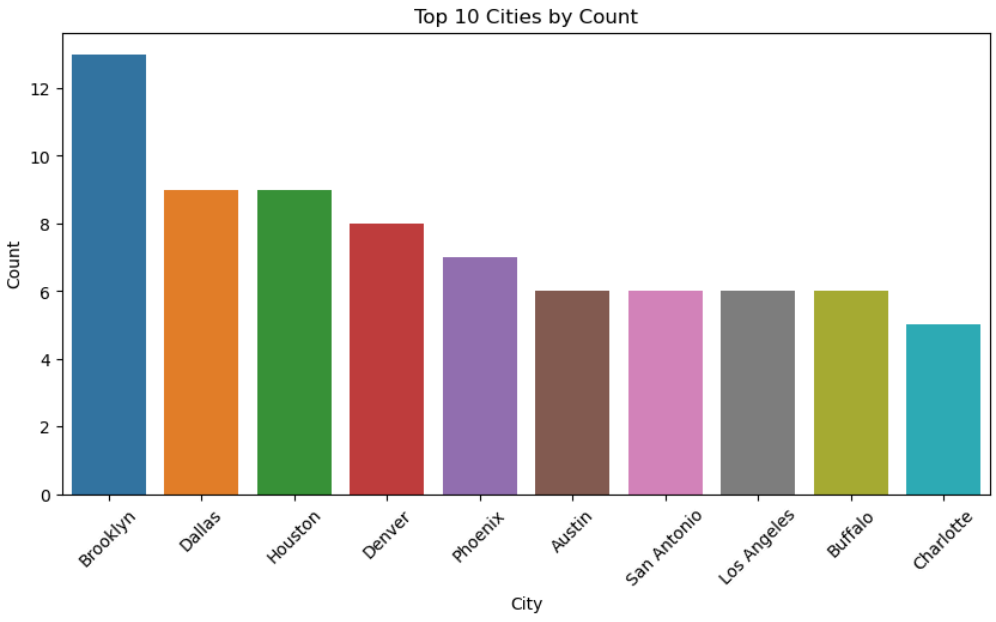


Figure 3.28 -Understanding the cities of residence of the customer base (For Business Case 03)

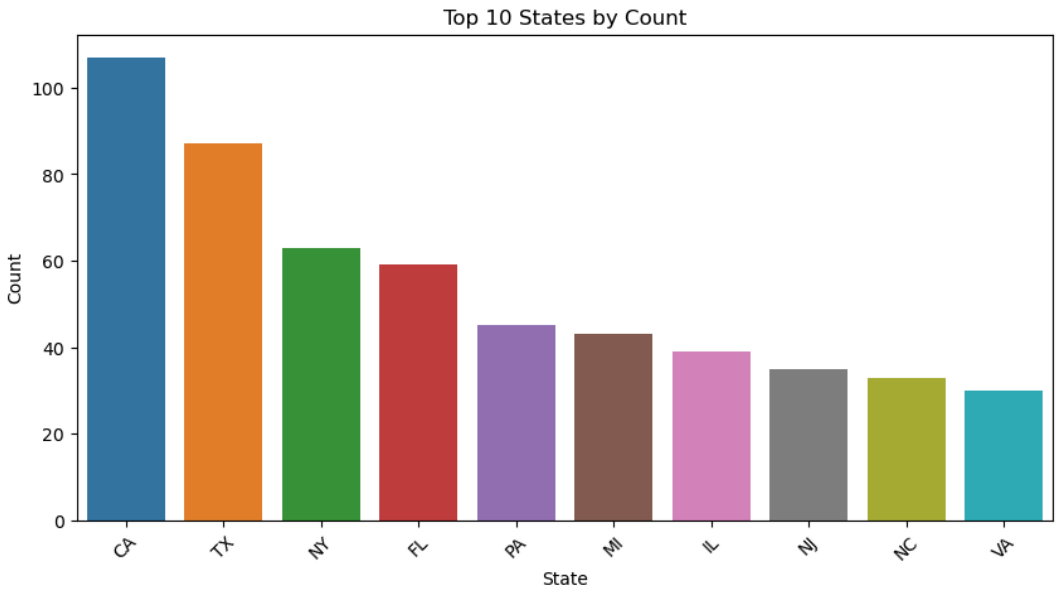


Figure 3.29 -Understanding the state of residence of the customer base (For Business Case 03)

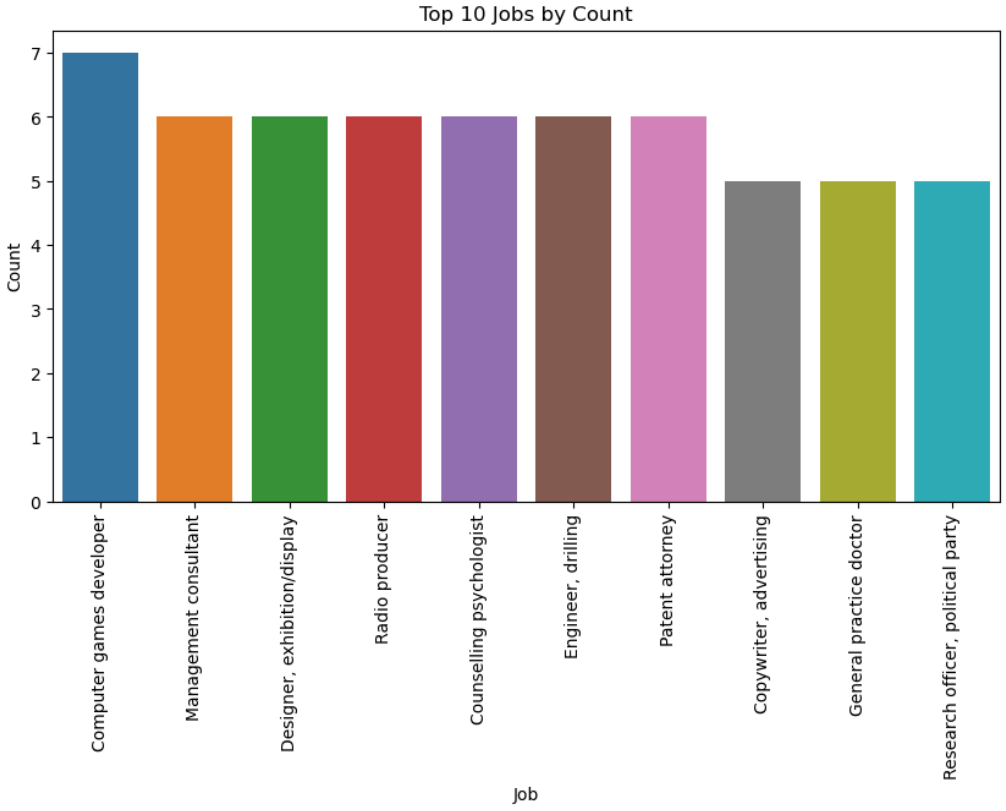


Figure 3.30 - Understanding the jobs of the customer base (For Business Case 03)

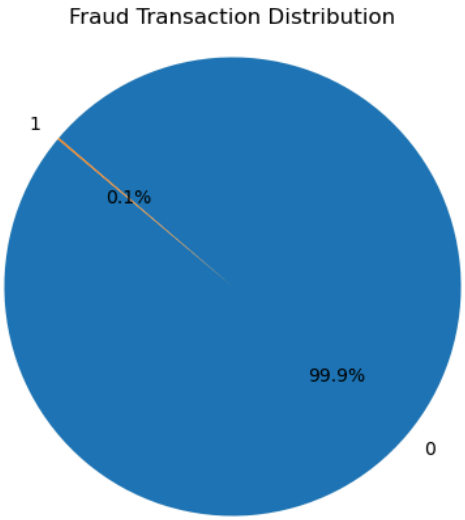


Figure 3.31 - Understanding the fraudulent transaction distribution of the dataset (For Business Case 03)

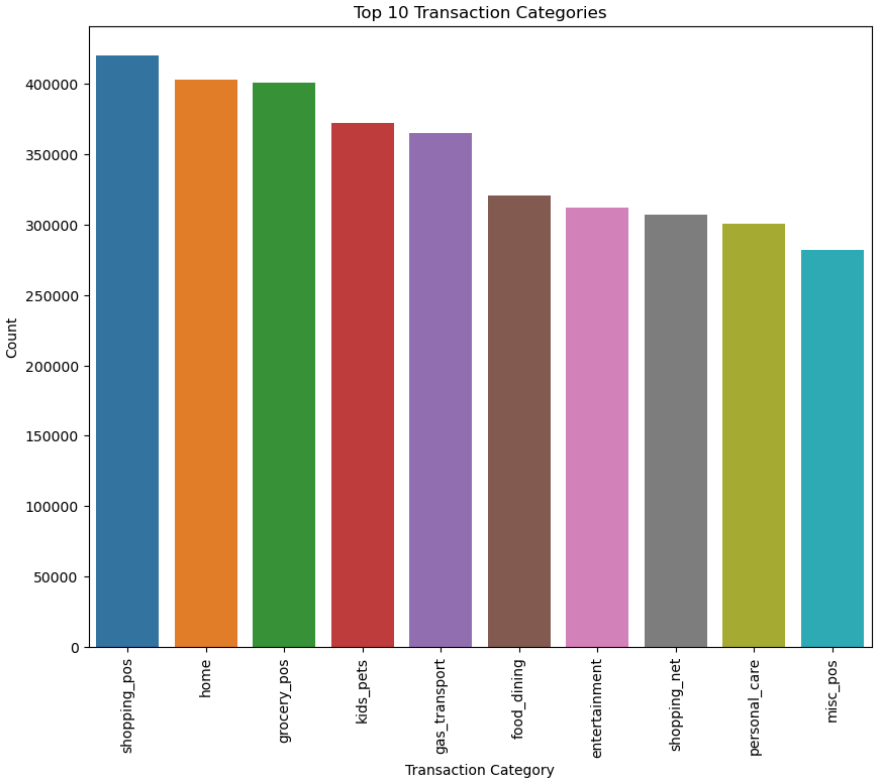


Figure 3.32 - Understanding the top transaction categories of the customer base (For Business Case 03)

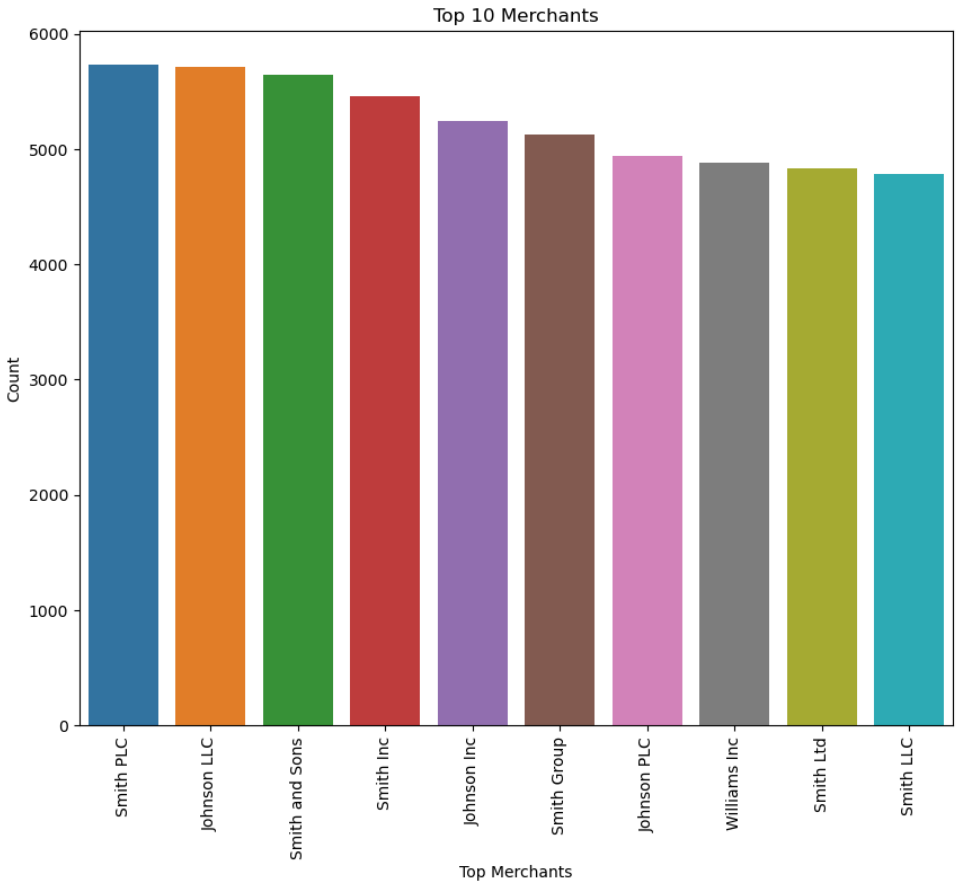


Figure 3.33 - Understanding the top merchants of the dataset (For Business Case 03)

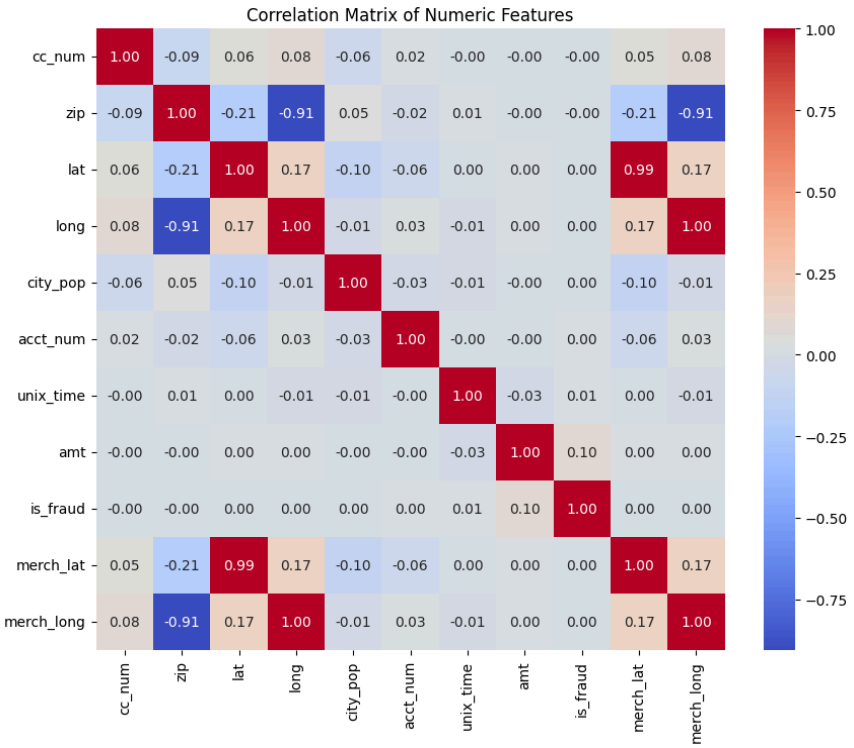


Figure 3.34 - understanding the relationship between numerical variables through correlation matrix (For Business Case 04)

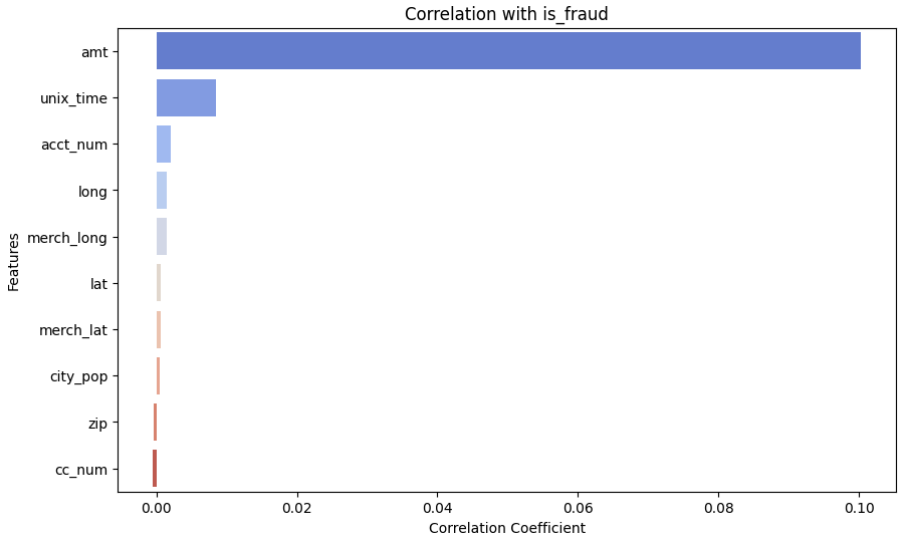


Figure 3.35 - understanding the significant features as per their correlation with fraud values utilized in the anomaly detection models (For Business Case 04)

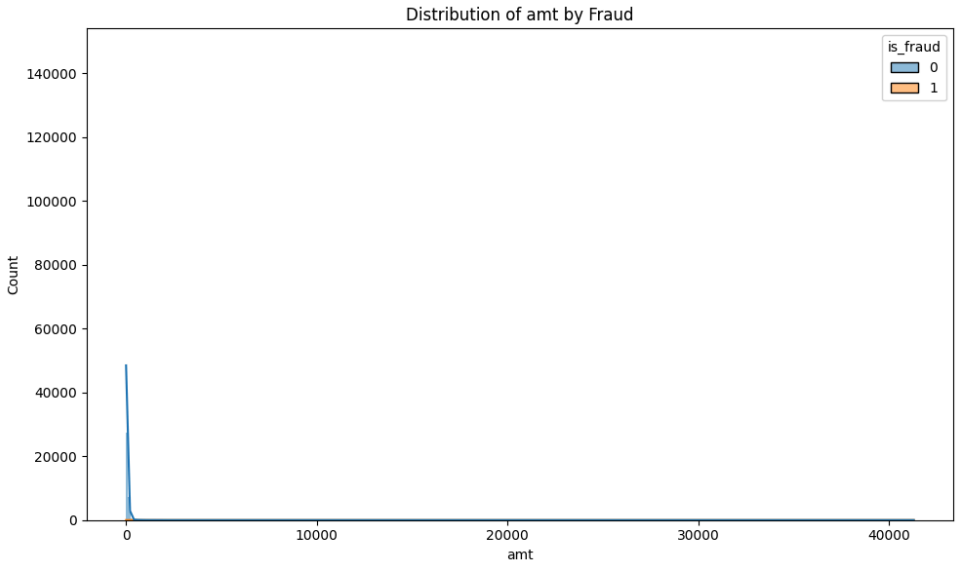


Figure 3.36 - understanding the distribution of fraud across amnt variable from the merged dataset (For Business Case 04)

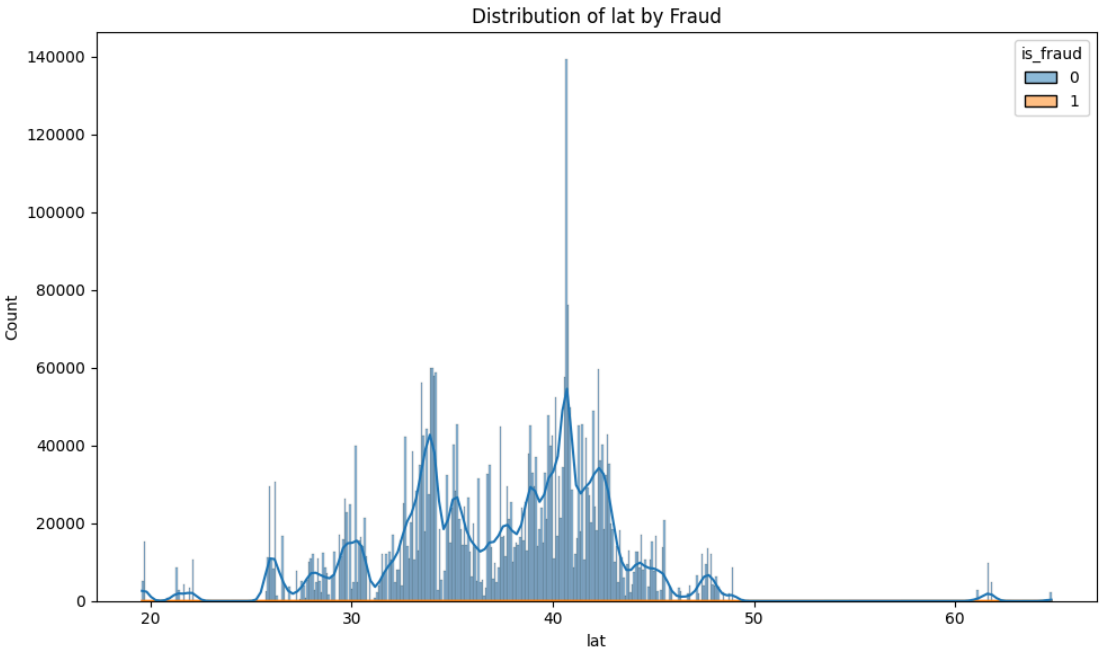


Figure 3.37 - understanding the distribution of fraud across lat variable from the merged dataset (For Business Case 04)

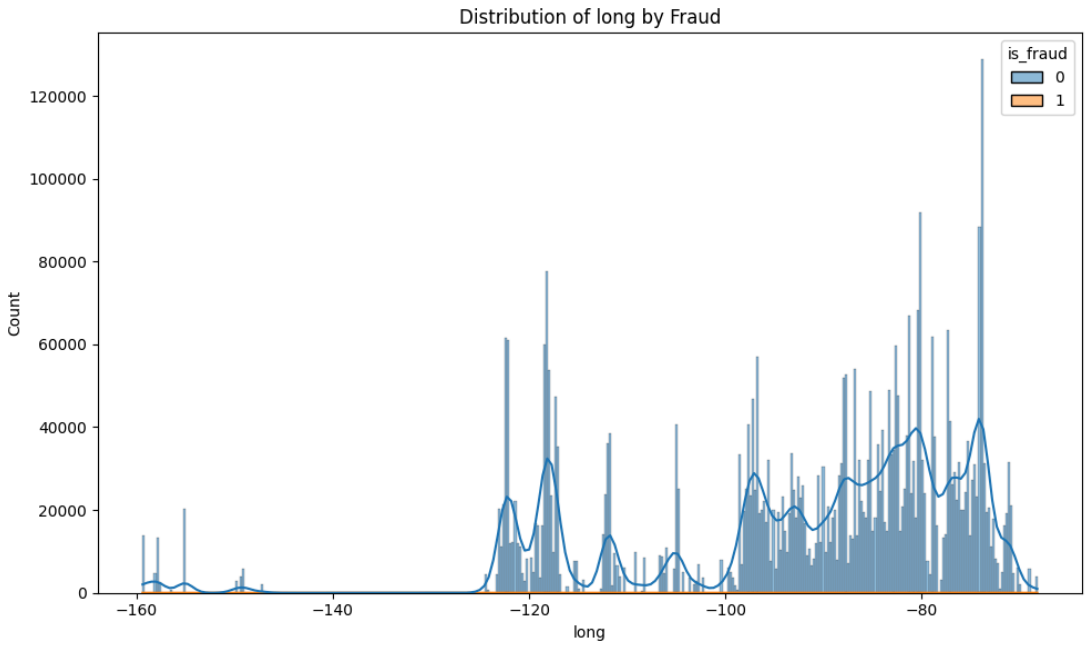


Figure 3.38 - understanding the distribution of fraud across long variable from the merged dataset (For Business Case 04)

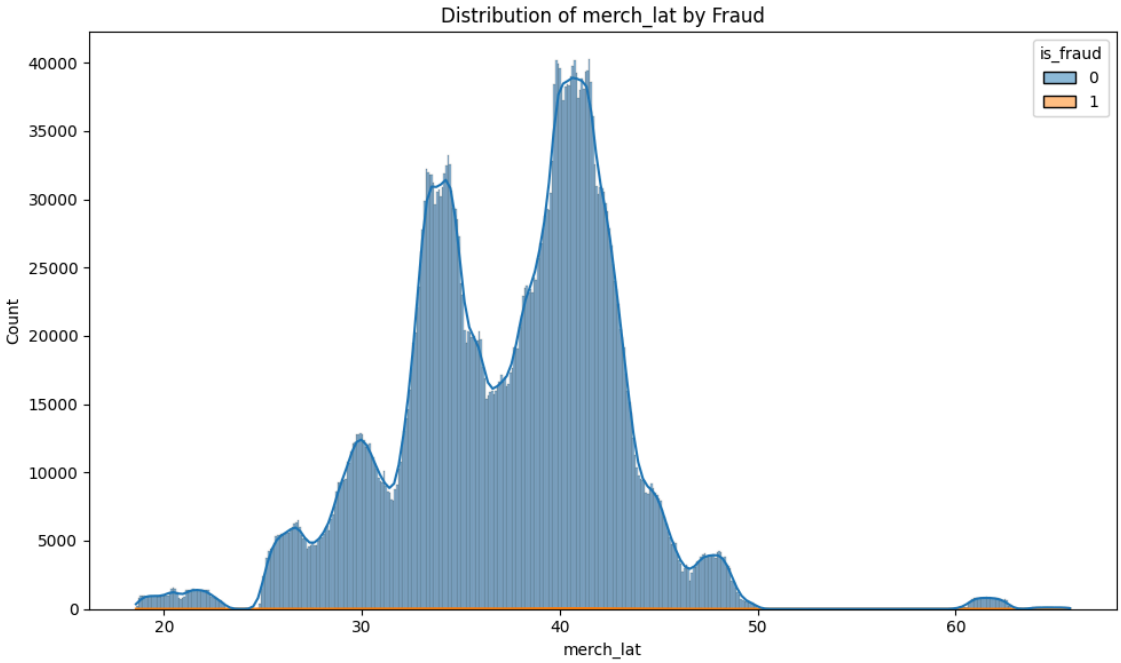


Figure 3.39 - understanding the distribution of fraud across merch_lat variable from the merged dataset (For Business Case 04)

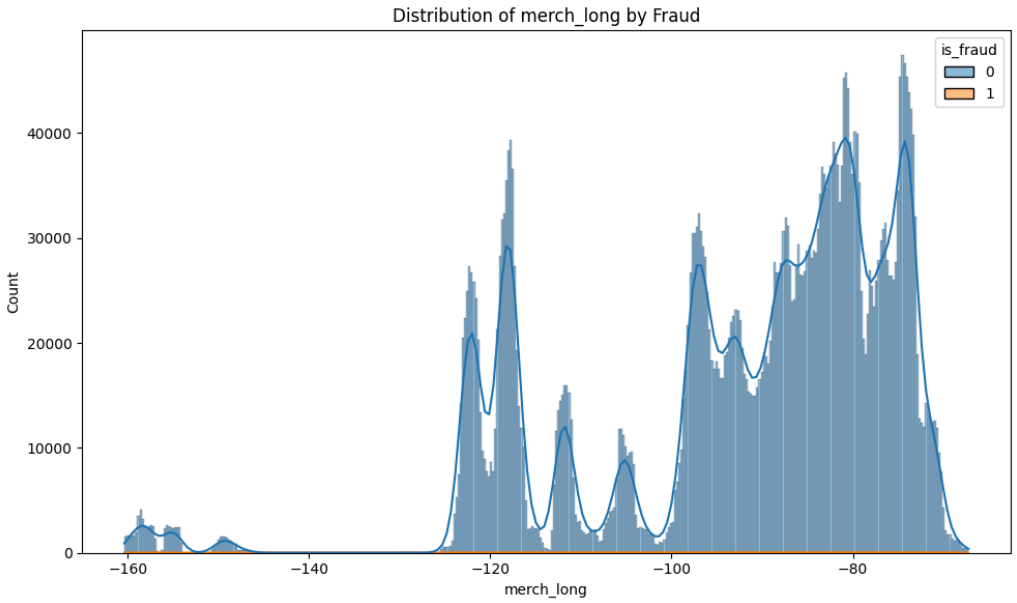


Figure 3.40 - understanding the distribution of fraud across merch_long variable from the merged dataset (For Business Case 04)

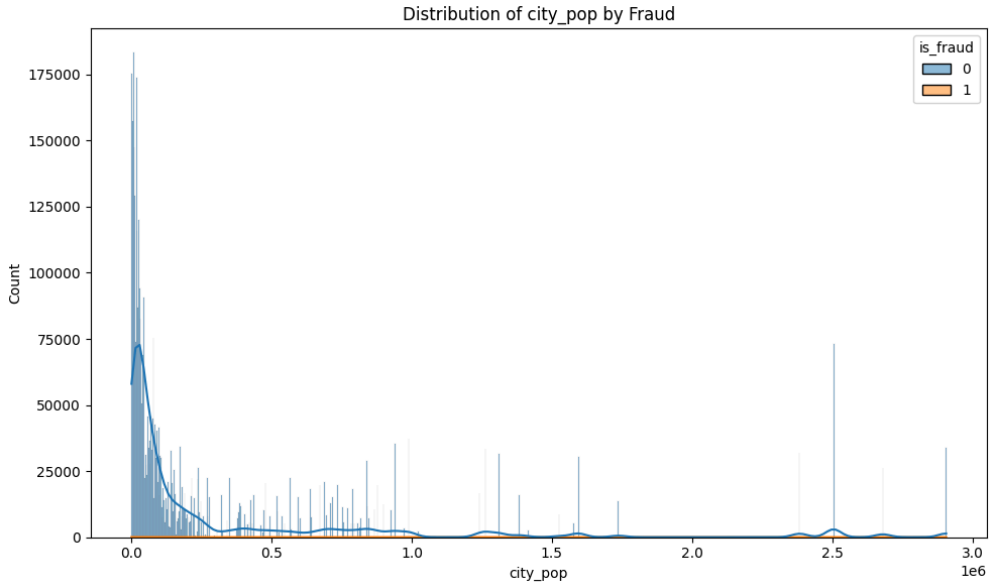


Figure 3.41 - understanding the distribution of fraud across city_pop variable from the merged dataset (For Business Case 04)

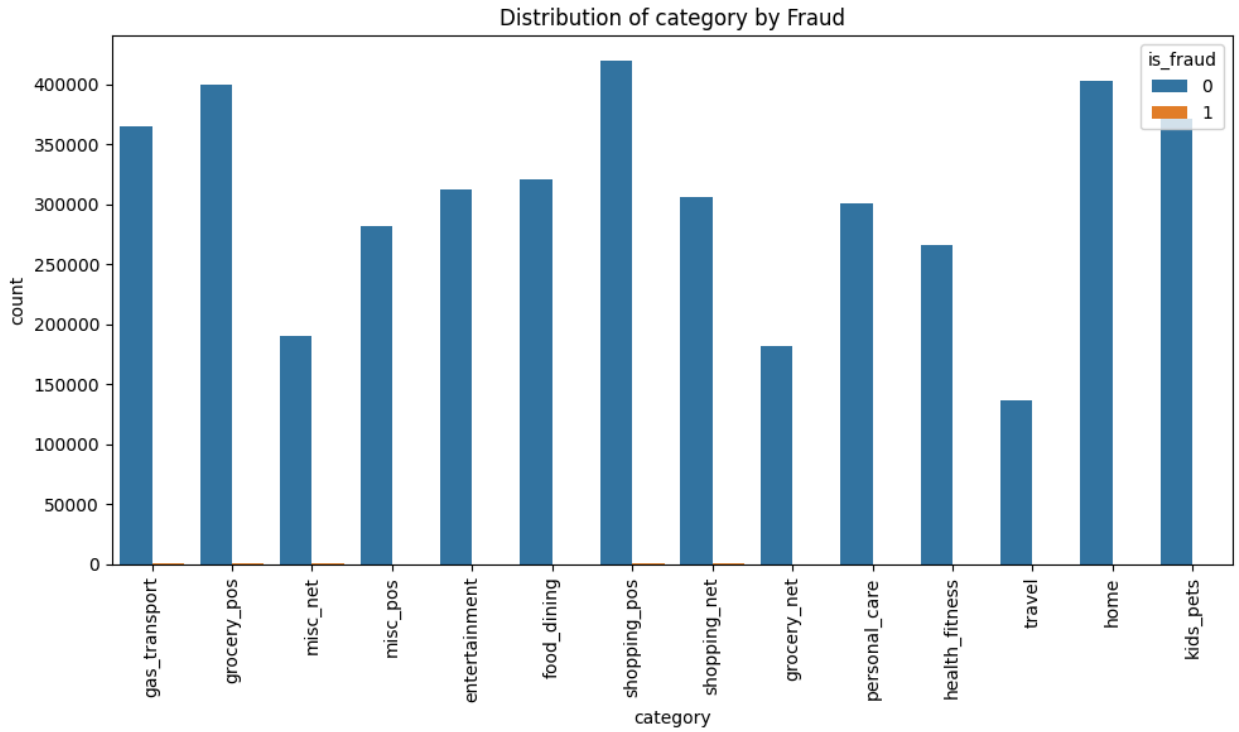


Figure 3.42 - understanding the distribution of fraud across categories of various transaction types variable from the merged dataset (For Business Case 04)

Data Preparation

Handling Missing Values and Duplicate Values

This involves imputing missing values with mean, median, or mode values, or using more sophisticated techniques such as predictive imputation. However, there were no missing values in either the transaction data or the customer data.

```
df_customer.isnull().sum()
```

```
ssn         0
cc_num      0
first       0
last        0
gender      0
street      0
city        0
state       0
zip         0
lat         0
long        0
city_pop    0
job         0
dob         0
acct_num    0
dtype: int64
```

```
duplicate_rows = df_customer[df_customer.duplicated()]
print(duplicate_rows)
```

```
Empty DataFrame
Columns: [ssn, cc_num, first, last, gender, street, city, state, zip, lat, long, city_pop, job, dob, acct_num]
Index: []
```

```
df_transaction.isnull().sum()
```

```
cc_num        0
acct_num      0
trans_num     0
unix_time     0
category      0
amt           0
is_fraud      0
merchant      0
merch_lat     0
merch_long    0
dtype: int64
```

```
duplicate_rows = df_transaction[df_transaction.duplicated]
print(duplicate_rows)
```

```
Empty DataFrame
Columns: [cc_num, acct_num, trans_num, unix_time, category, amt, is_fraud, merchant, merch_lat, merch_long]
Index: []
```

Removing Outliers

We should identify and remove outliers in the dataset that could negatively impact the performance of the classifier. Outliers can skew the distribution of the data and affect the decision boundaries learned by the model.

Dealing with Imbalanced Classes

Address class imbalance if present in the dataset. Class imbalance occurs when the number of instances in one class is significantly higher or lower than the other classes. Techniques such as oversampling, undersampling, or using algorithms that are robust to class imbalance can be applied.

Handling Categorical Variables

Encode categorical variables into numerical format, as most machine learning algorithms require numerical input. This could involve techniques such as one-hot encoding, label encoding, or target encoding.

Feature Scaling

Scale numerical features to a similar range to ensure that they contribute equally to the model's training process. Common scaling techniques include standardization (scaling features to have mean 0 and variance 1) or normalization (scaling features to a range between 0 and 1).

In business use case no 3, for this scenario, we used 2000 random rows from the entire merged dataset. This 2000 rows were selected randomly and then the standardization of the entire dataset was performed.

In business use case no 4, this was also implemented.

Removing Redundant Features

Identify and remove features that are redundant or highly correlated with other features in the dataset. Redundant features can increase model complexity without providing additional information, leading to overfitting.

For Business Use Case 01:

- Columns such as ‘job’, ‘city’, ‘street’,’state’,‘category’, ‘merchant’, ‘gender’, ‘dob’, ‘unixtime’, and ‘datetime’ were dropped after encoding to eliminate redundancy and noise.

- Similarly, columns such as ‘ssn’ and ‘trans_num’ were dropped because it had many unique values that could result in sparse data, which can negatively impact the model’s ability to generalize well on unseen data.

For Business Use Case 02:

- Drop Identifiers: Remove trans_num, cc_num, acct_num_x, acct_num_y, ssn, first, last, street, city, state, and zip, which contain personal information and don't forecast fraud.
- Address High Variance: Unique identifiers like cc_num, acct_num_x, unix_time, amt, zip, and city_pop may not be useful for fraud prediction.
- Moderate Variance: Day_of_week, month, merch_lat, merch_long, lat_customer, long_customer, lat_merchant, long_merchant, and distance_km may still be relevant.
- Eliminate Duplication: Remove superfluous characteristics like cc_num, acct_num_x, and acct_num_y, which identify fraud rather than anticipate it.
- Get rid of duplicates: If lat_merchant and long_merchant are merch_lat and merch_long, keep one pair of coordinates. Customer coordinate redundancy should be assessed.

Feature Engineering

For Business Case 01:

- **Binarization :** The ‘gender’ column was binarized with ‘M’ mapped to 1 and ‘F’ mapped to 0.
- **Age Calculation:** The ‘age’ column was derived from the ‘dob’ column for better predictive value.
- **Frequency Encoding** : It was applied to ‘job’ , ‘city’,’street’,’state’, ‘category’, and ‘merchant’,’first’, and ‘last’ due to their large number of unique values, making this a more suitable option than one-hot encoding or label encoding.

For Business Case 02:

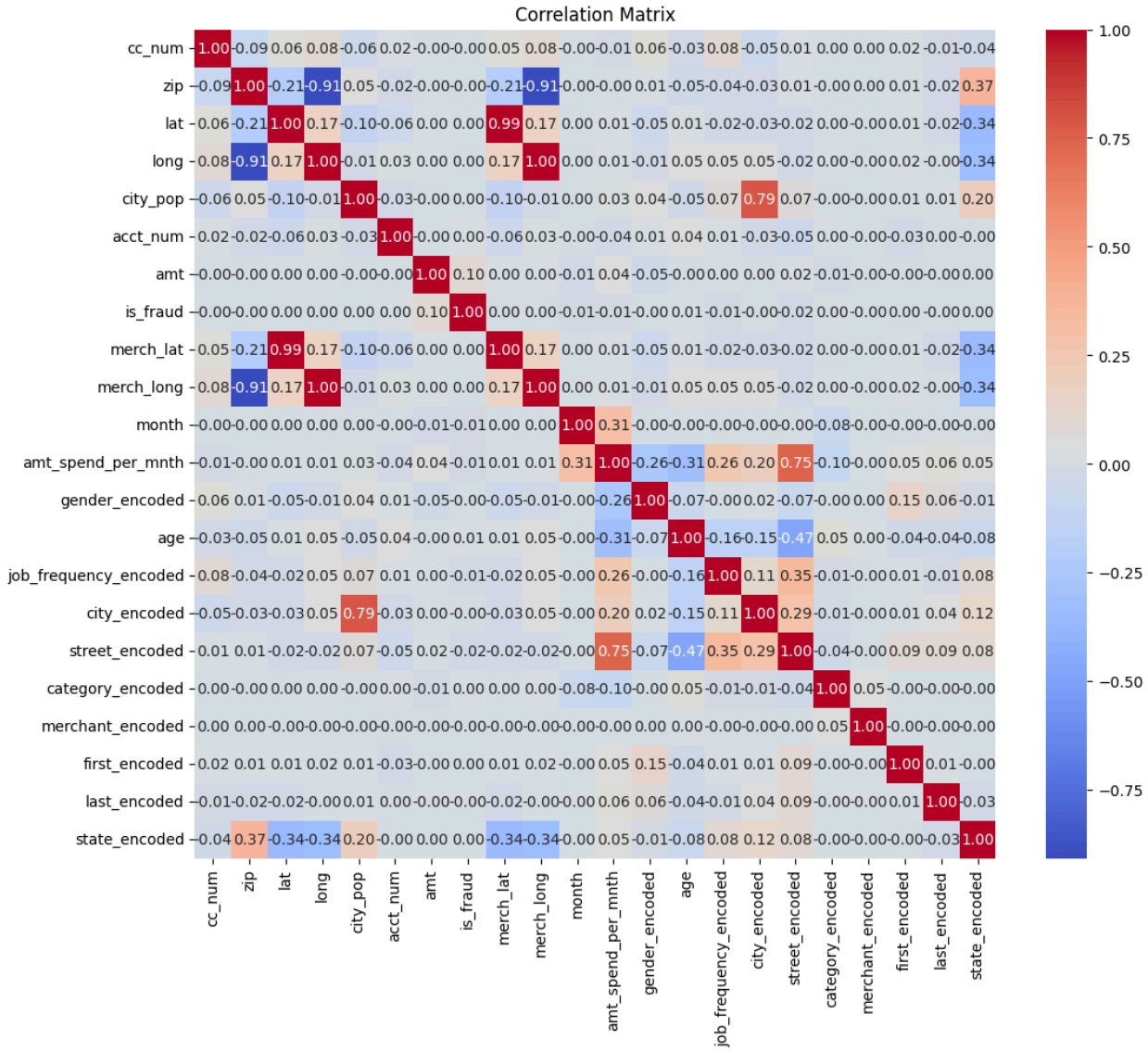
- Converted ‘unix time’ into ‘datetime’ and extracted ‘Hour of the day', 'Day of the week', and 'month' for each transaction and create new columns
- Distance between customer location and merchant transaction location:
- Some of the values in the ‘merch_long’ have ‘-’ in the end. Those were removed.
- Picked only the ‘lat’ and ‘long’ fields from the customer details and merged them with the transaction details using the ‘Account number’ field.
- And used the ‘merc_lat’ and ‘merch_long’ to find the distance and save it as a new field.

For Business Case 03:

- The columns gender_F and gender_M were generated through onehotkey encoding of the ‘gender’ column.
- The ‘year’ was generated by partitioning the year part from the ‘dob’ column.

Feature Selection

For Business Use Case 01:



The correlation analysis was performed to understand the relationships between the target variable and other attributes. Dimensionality reduction is a crucial step in data preprocessing, as

it helps improve model efficiency and performance by eliminating redundant or less relevant features. Based on the correlation analysis, feature selection was conducted accordingly.

The correlation analysis revealed that the amount spent per month has a **strong positive correlation with the street_encoded,month, job_frequency_encoded, and city_encoded**. Conversely, it showed a **negative correlation with age, gender_encoded, and category_encoded**. Consequently, these features were selected for data modeling to ensure efficient and accurate results. This approach leverages the importance of correlation analysis and dimensionality reduction to enhance model performance.

For Business Case 02:

We chose the following features based on the hypothesis derived from business understanding

- Category: From data POS and net transactions have more fraud count.
- Amt: High amount values could indicate fraud
    - Time, day, and month could be key factors
    - Time_hour
    - Time_day
    - Time_month
- Distance: The very long distance between the customer's place and the merchant's business could indicate fraud

For Business Use Case 3:

- To find the impactful columns to the clustering we used: 'city_pop', 'amt','year','is_fraud','gender_F', 'gender_M'
- There were some important columns like job, merchant, state which couldn't be used as they generated huge amounts of new columns after onehotey encoding.

For Business Use Case 04:

In order to investigate the significant variables for the anomaly detection models, it was essential to look into the numerical and categorical variables from the merged dataset.

Significant variables were selected in order to be included in the anomaly detection models from the combined dataset of 23 variables.

```
# Select relevant features
features = [
    'amt', 'lat', 'long', 'merch_lat', 'merch_long', 'city_pop',
    'category', 'merchant', 'state', 'city'
]
```

Figure 4.7 - code snippet of features/variables selected for modeling stages from the dataframe

Rationale for Feature Significance:

- Geographic Features (lat, long, merch_lat, merch_long, state, city):
    - These help in understanding the spatial patterns of transactions. Fraudulent transactions often deviate from the regular geographic patterns of a user's transactions.
- Behavioral Features (amt, category, merchant):
    - These capture the spending behavior of users. Fraudulent transactions typically deviate from the regular spending patterns in terms of amount, category, and merchant.
- Demographic Feature (city_pop):
    - Provides context on the likelihood of certain transaction types or amounts given the population size of the city.

Data Balancing

Unbalanced data has misclassified percentages. Most classes rule the data. Minorities are underrepresented. Not equal, class distribution is biased. Only 95% positive and 5% negative findings were used in our training dataset. Uneven categorization. Many factors can create an unequal classification predictive analysis task's class distribution. Domain properties and data sampling are to blame. The deadliest adversary of field data collection is observational or measurement inaccuracy. Mismeasurements result from poor measurement processes.

The model needs the training dataset to learn and classify. When trained with an imbalanced dataset (A:95; B:5), the model may become biased towards classifying everything as A, resulting in 90% accuracy but no ability to distinguish A from B. The loss function may not guide generalization training. Thus, a well-balanced training dataset helps the model comprehend what makes things A or B.

In addition to handling class imbalance, data balancing techniques such as SMOTE (Synthetic Minority Over-sampling Technique) can be used to generate synthetic samples for minority classes, thereby balancing the class distribution. Unbalanced datasets can bias machine learning models for fraud detection due to the high number of valid transactions. SMOTE (Synthetic Minority Over-sampling Technique) creates synthetic minority samples for a more balanced dataset. SMOTE converts a dataset with 9000 genuine and 1000 fraudulent transactions to 9000 of each. This balance helps the model learn from both classes equally, improving its fraud detection accuracy. Using SMOTE improves model fraud detection, making fraud detection systems more reliable.

[Make a balanced dataset]

In [96]:
```
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=42)
X_train_sm, y_train_sm = smote.fit_resample(X_train, y_train)

print('Original dataset shape:', y_train.value_counts())
print('Resampled dataset shape:', y_train_sm.value_counts())
```

```
Original dataset shape: 0    86097
1      134
Name: is_fraud, dtype: int64
Resampled dataset shape: 0    86097
1    86097
Name: is_fraud, dtype: int64
```

Figure 4.1 - Balanced dataset for "is_fraud" column in Business Use Case 02

Integrate Data

Integrating data requires the consolidation of various datasets or sources of information into one cohesive dataset for analysis. One possible task is to combine datasets that share common identifiers, such as keys or IDs. Another task is to add more data to an existing dataset. Additionally, you may need to gather data from various sources and consolidate it.

Implementing Before Data Splitting

- Ensures Consistency: Applying transformations before splitting the data ensures that the same preprocessing steps are applied to both the training and test sets. This is crucial for maintaining consistency in how the data is presented to the model.
- Prevents Data Leakage: If transformations like scaling or encoding are applied after data splitting, information from the test set could potentially influence the training process, leading to data leakage. This would result in overly optimistic performance estimates and poor generalization to new data.

Significance of these:

1. Data Integrity: Ensures that the data is in a suitable format for modeling, preventing errors during the training and prediction stages.
2. Model Performance: Improves the performance and reliability of machine learning models by providing them with well-preprocessed data.
3. Reproducibility: Facilitates a reproducible and systematic approach to data preprocessing, making it easier to replicate and validate results.

By implementing these preprocessing steps before data splitting and modeling, it can be ensured that machine learning workflow is robust, consistent, and free from common pitfalls such as data leakage.

Data Splitting

For Business Case 01: The dataset was split into training(60%), validation(20%), and testing(20%) sets to balance the large dataset size with the need for efficient model training. This split ensures that the model has sufficient data for training while retaining enough data for accurate validation and testing.

For Business Use Case 3: We used 2000 rows of the entire merged which we got randomly.

For Business Use Case 4: The data splitting was carried out train (80%) and test (20%) for effective implementation of anomaly detection models for the banking transactions dataset.

Modeling

Business Use Case 1

The same data preprocessing steps were applied across all three models to address the regression problem.

Multiple Linear Regression

- The statistical process for estimating the relationship between different variables is called regression analysis. Linear regression (LR) is a statistical method which models the relationship between dependent variable Y and one or more independent variable x. The function Y is called regression function. Linear regression (LR) models with more than one independent variable are called multiple linear models, as opposed to simple linear models, with one independent variable(Aleksandar, 2015).
- The Multiple Linear Regression was chosen for this project because it's easy to understand and interpret, making it straightforward to explain the relationship between independent variables and then target variable as well as it serves as a good starting point for comparing with more complex models.
- In this project, no hyperparameters were adjusted for the multiple regression model.

Elastic Net Regression

- Elastic Net is selected as the embedded method and it is the generalized form of LASSO and Ridge regression that has been adopted widely for high dimensional feature space regression problems(Amini, 2021).
- It was chosen because it helps in reducing model complexity and improving prediction accuracy by penalizing large coefficients.
- The hyperparameters for the Elastic Net model were tuned using randomized search CV, focusing on 'alpha' and 'l1_ratio'. The best parameters identified were {'alpha': 0.2058 and 'l1_ratio': 0.9699).

Gradient Boosting Regressor

- The Gradient Boosting Regressor (GBR) is another ensemble model that is an iterative collection of sequentially arranged tree models so as the next model learns from the error of the former model. This machine learning model makes predictions using "boosting" of the ensemble of the weak prediction models, often decision trees, to form a more robust model (Rao et al., 2019).

- It was chosen as it's known for its strong predictive performance by combining multiple weak learners to form a robust model.
- The specific hyperparameters were used : 'n_estimators = 100', 'learning_rate = 0.1', 'max_depth = 5','min_samples_split = 2', and 'min_samples_leaf=1'. These parameters control aspects like the number of boosting stages, the learning rate that adjusts the contribution of each tree, and the maximum depth of individual regression estimators. However, no hyperparameters tuning was conducted for Gradient boosting due to its lengthy training time.

Business Use Case 2

Logistic Regression:

One of the most prominent Supervised Learning methods is logistic regression. Based on independent variables, it predicts categorical dependent variables.

Logistic regression[2] predicts categorical dependent variable output. The consequence must be discrete or categorical. It can be Yes or No, 0 or 1, true or false, etc., but it delivers probability values between 0 and 1.

Model Rationale:

For interpretability and efficiency, Logistic Regression is used. It excels at binary classification issues like is_fraud because it shows how features like amt, gender, and distance_km affect fraud risk. It is useful for rapid, interpretable findings to learn how each attribute influences fraud probability..

Hyperparameter Tuning:

The code snippet optimizes hyperparameters for a logistic regression model using `GridSearchCV` to get optimal values for `C` and `penalty`. A grid of `C` values [0.001, 0.01, 0.1, 1, 10, 100] and penalties ['l1', 'l2'] is searched. After fitting grid search on training data, the best model is tested on validation data and presented using a confusion matrix, classification report, and ROC AUC score.

Code Snippet:

```python
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Train a Logistic Regression model
model = LogisticRegression(random_state=42)
model.fit(X_train_res, y_train_res)

# Make predictions on the validation set
y_pred = model.predict(X_val_res)

# Evaluate the model
accuracy = accuracy_score(y_val_res, y_pred)
precision = precision_score(y_val_res, y_pred)
recall = recall_score(y_val_res, y_pred)
f1 = f1_score(y_val_res, y_pred)

print("Validation Set Performance:")
print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1 Score: {f1:.4f}")
```

```python
from sklearn.metrics import roc_auc_score, roc_curve

# Calculate predictions for the validation set
y_pred_val = model.predict(X_val_res)

# Print confusion matrix and classification report for the validation set
print("Confusion Matrix for Validation Set:")
print(confusion_matrix(y_val_res, y_pred_val))
print("\nClassification Report for Validation Set:")
print(classification_report(y_val_res, y_pred_val))
print("\nROC AUC Score for Validation Set:", roc_auc_score(y_val_res, y_pred_val))

# Plot ROC curve for the validation set
fpr_val, tpr_val, _ = roc_curve(y_val_res, model.predict_proba(X_val_res)[:,1])
plt.figure(figsize=(8, 6))
plt.plot(fpr_val, tpr_val, marker='.')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve for Validation Set')
plt.show()

```

```python
from sklearn.model_selection import GridSearchCV

# Define the hyperparameters grid
param_grid = {
    'C': [0.001, 0.01, 0.1, 1, 10, 100],  # Regularization parameter
    'penalty': ['l1', 'l2']  # Regularization type
}

# Create a grid search object
grid_search = GridSearchCV(LogisticRegression(random_state=42), param_grid, cv=5, scoring='f1')

# Perform grid search on the training data
grid_search.fit(X_train_res, y_train_res)

# Get the best hyperparameters
best_params = grid_search.best_params_
print("Best Hyperparameters:", best_params)

# Get the best model
best_model = grid_search.best_estimator_

# Make predictions on the validation set using the best model
y_pred_val_best = best_model.predict(X_val_res)

# Print confusion matrix and classification report for the validation set using the best model
print("\nConfusion Matrix for Validation Set (Best Model):")
print(confusion_matrix(y_val_res, y_pred_val_best))
print("\nClassification Report for Validation Set (Best Model):")
print(classification_report(y_val_res, y_pred_val_best))
print("\nROC AUC Score for Validation Set (Best Model):", roc_auc_score(y_val_res, y_pred_val_best))
```

Decision Trees :

Model Rationale: Decision Trees are easy to visualize and interpret. They have explicit decision rules based on category, hour_of_day, and day_of_week and can capture nonlinear interactions to discover anomalous transaction patterns. This model helps explain decision-making and feature interactions.

Hyperparameter Tuning: GridSearchCV optimises Decision Tree Classifier hyperparameters max_depth, min_samples_split, and min_samples_leaf. A grid of max_depth [None, 10, 20, 30, 40, 50], min_samples_split [2, 5, 10], and min_samples_leaf [1, 2, 4] is searched. Following grid search on training data, the best model is tested on validation data and displayed using a confusion matrix, classification report, and ROC AUC score.

Naive Bayes:

Naive Bayes: Naive Bayes classifiers employ probabilistic machine learning to classify. It uses Bayes' theorem to estimate an event's probability based on prior knowledge of associated conditions. Naive Bayes performs well in many real-world situations, especially with high-dimensional data, despite its simplicity.

Model Rationale: Naive Bayes is chosen for its simplicity, efficiency, and high-dimensional dataset performance. Each pair of features is assumed to be independent in Bayes' theorem. This model works well when predictive factors are conditionally independent of class. Naive Bayes is notable for its text classification and spam detection abilities despite its simplicity.

Hyperparameter Tuning: GridSearchCV tweaks hyperparameters to optimize Naive Bayes classifier performance. Var_smoothing, which governs the addition of the biggest variance of all features to variances for computation stability, is the crucial hyperparameter. The grid search investigates var_smoothing values [1e-9, 1e-8, 1e-7, 1e-6, 1e-5]. The F1 score determines the best parameter setting using 5-fold cross-validation.

Code Snippet:

```
# Train a Naive Bayes model
nb_model = GaussianNB()
nb_model.fit(X_train_res, y_train_res)

# Make predictions on the validation set
y_pred_nb = nb_model.predict(X_val_res)

# Evaluate the model
accuracy_nb = accuracy_score(y_val_res, y_pred_nb)
precision_nb = precision_score(y_val_res, y_pred_nb)
recall_nb = recall_score(y_val_res, y_pred_nb)
f1_nb = f1_score(y_val_res, y_pred_nb)

print("Validation Set Performance (Naive Bayes):")
print(f"Accuracy: {accuracy_nb:.4f}")
print(f"Precision: {precision_nb:.4f}")
print(f"Recall: {recall_nb:.4f}")
print(f"F1 Score: {f1_nb:.4f}")
```

```python
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score, roc_curve
import matplotlib.pyplot as plt

# Train a Naive Bayes model
nb_model = GaussianNB()
nb_model.fit(X_train_res, y_train_res)

# Calculate predictions for the validation set
y_pred_val_nb = nb_model.predict(X_val_res)

# Print confusion matrix and classification report for the validation set
print("Confusion Matrix for Validation Set:")
print(confusion_matrix(y_val_res, y_pred_val_nb))
print("\nClassification Report for Validation Set:")
print(classification_report(y_val_res, y_pred_val_nb))
print("\nROC AUC Score for Validation Set:", roc_auc_score(y_val_res, y_pred_val_nb))

# Plot ROC curve for the validation set
fpr_val_nb, tpr_val_nb, _ = roc_curve(y_val_res, nb_model.predict_proba(X_val_res)[:,1])
plt.figure(figsize=(8, 6))
plt.plot(fpr_val_nb, tpr_val_nb, marker='.')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve for Validation Set (Naive Bayes)')
plt.show()
```

```python
from sklearn.model_selection import GridSearchCV
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score

# Define the hyperparameters grid
param_grid = {
    'var_smoothing': [1e-9, 1e-8, 1e-7, 1e-6, 1e-5]  # Smoothing parameter
}

# Create a grid search object
grid_search = GridSearchCV(GaussianNB(), param_grid, cv=5, scoring='f1')

# Perform grid search on the training data
grid_search.fit(X_train_res, y_train_res)

# Get the best hyperparameters
best_params = grid_search.best_params_
print("Best Hyperparameters:", best_params)

# Get the best model
best_model = grid_search.best_estimator_

# Make predictions on the validation set using the best model
y_pred_val_best = best_model.predict(X_val_res)

# Print confusion matrix and classification report for the validation set using the best model
print("\nConfusion Matrix for Validation Set (Best Model):")
print(confusion_matrix(y_val_res, y_pred_val_best))
print("\nClassification Report for Validation Set (Best Model):")
print(classification_report(y_val_res, y_pred_val_best))
print("\nROC AUC Score for Validation Set (Best Model):", roc_auc_score(y_val_res, y_pred_val_best))
```

Random Forests: (Baseline Model)

Random Forest

"The Random Forest classifier[6] is designed to enhance the predictive accuracy of a dataset by utilizing multiple decision trees on different subsets of the data and averaging their results." Similar to a market research analyst, the random forest algorithm doesn't rely on just one decision tree. Instead, it combines the predictions from multiple trees and uses the majority vote to determine the final output. Having a larger number of trees in the forest ensures improved accuracy and mitigates the issue of overfitting.

Model Rationale:

Robust Random Forests use several decision trees to improve accuracy and prevent overfitting. They accurately estimate feature importance, identifying fraud detection indicators like amt and distance_km. Robustness and generalization to fresh data make this model useful.

Support Vector Machines (SVM):

The support vector machine[4] method is a statistical classification strategy that optimizes the hyperplane separation margin between cases. It's a non-probabilistic binary linear classifier that can effectively differentiate between classes by a large margin. It's a powerful classifier that can handle any size feature vector. A support vector machine (SVM) is a regression and classification algorithm that uses supervised machine learning. It is, however, most commonly utilized when there are problems with classification.

Model Rationale

SVMs work well in high-dimensional spaces, making them suited for your dataset with amt, category, merchant, merch_lat, merch_long, gender, lat_merchant, long_merchant, job, hour_of_day, day_of_week, month, lat_customer, long_customer, and distance_km. SVMs can increase generalization and capture the appropriate decision boundary in high-dimensional feature space by optimizing the margin between is_fraud classes.

Optimizing Hyperparameters: The code snippet optimizes SVM classifier hyperparameters using `RandomizedSearchCV` to determine optimal values for `C`, `kernel`, and `gamma`. Various `C` values (np.logspace(-3, 3, 10)`, `kernel` options ('linear', 'poly', 'rbf','sigmoid'), and `gamma` choices ('scale', 'auto') are searched After randomizing the training data, the best model is tested on validation data. The confusion matrix, classification report, and ROC AUC score show the results. The model's validation set performance is also represented using a ROC curve.

K-Nearest Neighbors (KNN):

K-nearest neighbors (K-NN) **[5]** is a simple and widely used supervised learning method in machine learning. The algorithm functions based on the premise that new data bears resemblance to previous data, and consequently assigns the new example to the nearest available category. The K-NN algorithm preserves all data and classifies new data points by measuring their similarity to existing ones. It efficiently categorizes fresh data into appropriate classes, primarily used for jobs involving classification.

Model Rationale: KNN predicts using the nearest neighbor majority class and is easy to construct and comprehend. The non-parametric model does not assume any data distribution, which is useful for heterogeneous data with categorical (category, gender, job), continuous (amt, distance_km), and geospatial features. This model captures local data patterns well and is effective for smaller datasets.

Hyperparameter tuning : The code snippet optimizes K-Nearest Neighbors (KNN) classifier hyperparameters using GridSearchCV to get optimal values for `n_neighbors`, `weights`, and `metric`. The search is conducted using a grid of `n_neighbors` [3, 5, 7, 9, 11], weights ['uniform', 'distance'], and metrics ['euclidean','manhattan','minkowski']. After fitting the grid search on training data, the best model is tested on validation data using a confusion matrix, classification report, and ROC AUC score. The validation set's ROC curve is plotted to show model performance.

Neural Networks:

Neural Networks:

Neural networks, especially deep learning architectures, capture complicated data patterns well. Layers of neurons process input data and learn hierarchical representations. They work well for large-data, feature-learning fraud detection.

Model Rationale: Flexible neural networks can describe complex non-linear feature connections. Your complex dataset benefits from their ability to handle large datasets with multiple features. Neural Networks may discover complex patterns in data where amt, distance_km, hour_of_day, and day_of_week interact. This model can detect fraud in huge, complicated datasets with non-linear correlations.

Hyperparameter tuning : The code snippet optimizes MLP Classifier hyperparameters using GridSearchCV to determine optimal settings for hidden_layer_sizes, activation, solver, and alpha. A grid with hidden layer sizes (100, 50, 50, 50), activation functions (logistic, tanh, relu), solver options (adam, sgd), and alpha values (0.0001, 0.001, 0.01), is searched. After fitting the grid search on training data, the best model is tested on validation data using a confusion matrix, classification report, and ROC AUC score.

Business Use Case 3

K-Means Clustering

K-Means clustering partitions n observations into k clusters where each observation belongs to the cluster with the nearest mean. The model is simple, efficient, and effective for large datasets. It's ideal for customer segmentation due to its ability to create distinct groups based on spending behaviors.

The hyperparameters are the number of clusters (k), initialization method(k-means++), and number of initializations (n_init). We determined the optimal number of clusters using the Elbow Method.

The elbow method is a way to figure out the best number of groups (clusters) to split the data into. The method works by trying different numbers of clusters and seeing how well the data fit in each cluster. We then plot a graph showing how well the data fit compared to the number of clusters we tried. This graph often looks like an elbow. The ideal number of clusters is at the bend of the elbow, where adding more clusters doesn't make things much better.

We made sure that we performed the Standardization using StandardScaler such that all the necessary features in the model had a mean of 0 and a standard deviation of 1. Finally, trained the model on the preprocessed data, iteratively adjusting the number of clusters (k) to find the optimal clustering solution.

Mean Shift Clustering

This non-parametric technique excels at identifying clusters with no predefined number of groups. It iteratively shifts data points towards denser regions within the data space, making it suitable for customer data with potentially complex spending patterns and varying cluster sizes.

The key hyperparameter is the bandwidth, which determines the size of the area used to assess density around each data point.

Similar to K-Means clustering, data standardization (using StandardScaler) ensures all features have equal weight during clustering.

After preprocessing, the Mean Shift algorithm iteratively shifts data points towards denser areas, identifying cluster centers.

Agglomerative Clustering

This hierarchical clustering method starts with each data point as a separate cluster and progressively merges the closest pairs, building a tree-like structure (dendrogram) that reveals the hierarchical organization of the data. This approach is ideal for uncovering nested clusters within customer segments.

A key hyperparameter was considered - distance threshold (maximum distance for cluster merging).

Similar to Mean Shift, data standardization ensures consistent feature scaling. The same set of features representing spending behavior is utilized.

The algorithm starts with each data point as a separate cluster and iteratively merges the closest pairs based on the chosen linkage criterion. This process continues until the desired number of clusters or a specified distance threshold is reached. The resulting dendrogram aids in determining the optimal number of clusters and understanding the hierarchical structure of the customer data.

The three clustering algorithms (K-Means, Mean Shift, and Agglomerative Clustering) were used to segment customers based on certain characteristics. These algorithms were selected for their ability to identify distinct customer groups, their ease of interpretation, and their flexibility in handling different data structures. Hyperparameter tuning involved techniques like the Elbow Method, and dendrogram analysis to find the optimal model parameters. The training of the model focused on standardizing the data, creating relevant features, and ensuring robust cluster formation through iterative tuning and validation.

Each approach provided unique insights into customer segmentation, allowing the marketing team to tailor their strategies effectively.

Business Use Case 4

Isolation forest

Isolation Forest is an unsupervised learning algorithm specifically designed for anomaly detection. It operates on the principle that anomalies are 'few and different,' making them more susceptible to isolation by randomly partitioning the data.

Key Hyperparameters:

- n_estimators: The number of base estimators (trees) in the ensemble. This determines how many trees are used to build the forest. A typical default value is 100.
- max_samples: The number of samples to draw from the dataset to train each base estimator. This can be set to a specific number or as a fraction of the total number of samples (e.g., 256 or 0.25).
- contamination: The proportion of the dataset expected to contain anomalies. This is a critical parameter for adjusting the sensitivity of the model. In our case, it was set to 0.1, meaning we expected 10% of the data to be anomalies.
- max_features: The number of features to draw from the dataset to train each base estimator. This can be set to a specific number or as a fraction of the total number of features.
- random_state: A seed used by the random number generator to ensure reproducibility of results.

Preprocessing and Feature Engineering

Feature Selection:

- Selected features: 'amt', 'lat', 'long', 'merch_lat', 'merch_long', 'city_pop', 'category', 'merchant', 'state', 'city'.
- Excluded features: 'unix_time', 'hour', and 'day_of_week'.

Preprocessing:

- Numerical Features: Scaling: StandardScaler was used to standardize the numerical features ('amt', 'lat', 'long', 'merch_lat', 'merch_long', 'city_pop') by removing the mean and scaling to unit variance. This ensures that all numerical features contribute equally to the distance metric used in the Isolation Forest algorithm.

- Categorical Features: Encoding: OneHotEncoder was used to transform categorical features ('category', 'merchant', 'state', 'city') into a binary format. Each category level was converted into a separate binary column, which allows the model to handle categorical data effectively. To avoid multicollinearity, the first category was dropped using the drop='first' parameter.
- Pipeline: A pipeline was constructed to streamline the preprocessing and model training processes. The ColumnTransformer was used to apply the appropriate transformations to the numerical and categorical features within the pipeline.

Pipeline Construction:A Pipeline was constructed with two main steps: Preprocessor: This step included the ColumnTransformer to apply StandardScaler to numerical features and OneHotEncoder to categorical features. Model: The IsolationForest model was added as the final step in the pipeline.

Model Training: The fit method of the pipeline was used to train the model on the selected features from the dataset. The pipeline ensures that preprocessing steps are applied consistently during both training and prediction phases.

Handling Imbalanced Data: Isolation Forest inherently handles imbalanced data by focusing on the isolation of individual samples. By setting the contamination parameter to 0.1, the model is configured to expect that approximately 10% of the data points are anomalies. This parameter helps the algorithm determine the threshold for classifying a data point as an anomaly.

Anomaly Detection: After training, the decision_function method was used to compute the anomaly scores for each data point. The predict method was then used to classify each data point as either an anomaly (-1) or normal (1). These labels were then mapped to binary format (1 for anomalies and 0 for normal data points) for easier interpretation.

Results and Visualization

The data with anomaly labels was printed to examine the results.

K- means clustering

K-Means is an unsupervised learning algorithm used for clustering data into K distinct groups based on feature similarity. Each data point is assigned to the cluster with the nearest mean.

Key Hyperparameters:

- n_clusters: The number of clusters to form. For anomaly detection, n_clusters is typically set to 2 to separate normal data points from anomalies.
- init: Method for initialization. k-means++ is often used as it helps spread out the initial cluster centers.
- max_iter: Maximum number of iterations the algorithm will run for a single initialization (default is 300).
- n_init: Number of different initializations to perform. The final result will be the best output from these initializations (default is 10).
- random_state: A seed used by the random number generator to ensure reproducibility of results.

Preprocessing and Feature Engineering

Feature Selection:

- Selected features: 'amt', 'lat', 'long', 'merch_lat', 'merch_long', 'city_pop', 'category', 'merchant', 'state', 'city'.
- Excluded features: 'unix_time', 'hour', and 'day_of_week'.

Preprocessing:

- Numerical Features: Scaling: StandardScaler was used to standardize the numerical features ('amt', 'lat', 'long', 'merch_lat', 'merch_long', 'city_pop'). Standardization removes the mean and scales each feature to unit variance, ensuring that all numerical features contribute equally to the clustering process.
- Categorical Features: Encoding: OneHotEncoder was used to transform categorical features ('category', 'merchant', 'state', 'city') into a binary format. Each level of a categorical feature was converted into a separate binary column, allowing the algorithm to handle categorical data effectively.

Pipeline: A pipeline was constructed to streamline the preprocessing and model training processes. The ColumnTransformer was used to apply the appropriate transformations to the numerical and categorical features within the pipeline.

Pipeline Construction: A ColumnTransformer was created to preprocess the data. StandardScaler was applied to numerical features. OneHotEncoder was applied to categorical features. The transformed data was then used to fit the K-Means model.

Splitting the Data: The dataset was split into training and testing sets using train_test_split, with 80% of the data used for training and 20% for testing.

Model Training: The K-Means model was trained on the preprocessed training data. The number of clusters was set to 2 to identify normal data points and anomalies.

Predicting Cluster Labels: The model predicted cluster labels for both the training and testing data.

Handling Imbalanced Data: Since K-Means does not inherently handle imbalanced data, the approach was to determine the majority cluster label. The majority cluster label (the one with the most data points) was assumed to represent the normal data points, while the minority cluster label was assumed to represent anomalies.

Anomaly Detection: Anomalies were identified by comparing each data point's cluster label to the majority label. Data points that did not belong to the majority cluster were classified as anomalies.

Local Outlier Factor

The Local Outlier Factor (LOF) algorithm is an unsupervised anomaly detection method that identifies anomalies by measuring the local density deviation of a given data point with respect to its neighbors. Anomalies are points that have a significantly lower density compared to their neighbors.

Key Hyperparameters:

n_neighbors: The number of neighbors to use for calculating the local density. A common choice is 20, but this can be adjusted based on the dataset.

algorithm: The algorithm used to compute the nearest neighbors. Options include ball_tree, kd_tree, or auto (default).

leaf_size: Leaf size passed to the underlying tree algorithms. It affects the speed of the construction and query, as well as the memory required to store the tree.

metric: The distance metric used to calculate the neighbors. Common choices include euclidean, manhattan, or minkowski.

novelty: When set to True, allows the model to be used for outlier detection on new, unseen data.

Preprocessing and Feature Engineering

Feature Selection: Selected features: 'amt', 'lat', 'long', 'merch_lat', 'merch_long', 'city_pop', 'category', 'merchant', 'state', 'city'. Excluded features: 'unix_time', 'hour', and 'day_of_week'.

Preprocessing:

Numerical Features: Scaling: StandardScaler was used to standardize the numerical features ('amt', 'lat', 'long', 'merch_lat', 'merch_long', 'city_pop'). Standardization removes the mean and scales to unit variance, ensuring that all numerical features contribute equally to the anomaly detection process.

Categorical Features: Encoding: OneHotEncoder was used to transform categorical features ('category', 'merchant', 'state', 'city') into a binary format. Each level of a categorical feature was converted into a separate binary column.

Dimensionality Reduction: TruncatedSVD: Applied to reduce the dimensionality of the preprocessed features. This helps in mitigating the curse of dimensionality and enhances the performance of the LOF model. For this example, the number of components was set to 10.

Pipeline: A pipeline was constructed to streamline the preprocessing, dimensionality reduction, and model training processes.

Training Process

Pipeline Construction: A ColumnTransformer was created to preprocess the data. StandardScaler was applied to numerical features. OneHotEncoder was applied to categorical features. TruncatedSVD was used to reduce the dimensionality of the preprocessed features.

Data Splitting: The dataset was split into training and testing sets using train_test_split, with 80% of the data used for training and 20% for testing.

Model Training: The LocalOutlierFactor model was trained on the reduced training data. The novelty parameter was set to True to allow the model to be used for outlier detection on new, unseen data.

Predicting Anomalies: The model predicted anomalies on the testing data. Predictions were made using the predict method, which outputs -1 for anomalies and 1 for normal data points. Anomalies were identified by mapping -1 (anomaly) to True and 1 (normal) to False.

Evaluation

Evaluation Metrics

Business Use Case 1

The evaluation metric used to assess the model's performance is root mean squared error **(RMSE)** for multiple linear regression, elastic net regression, and gradient boosting regressor. RMSE was chosen because it can effectively measure the average magnitude of the errors between predicted and actual values, providing an indication of the model's accuracy.

Business Use Case 2

The following metrics are essential for evaluating your fraud detection project's models:

Precision: Precision is the percentage of expected fraud cases correctly discovered.

Importance: High precision in fraud detection assures that most flagged transactions are fraudulent, eliminating costly and disruptive false positives.

Recall: Recall is the percentage of real fraud cases accurately identified.

High recall helps identify most fraudulent transactions, reducing false negatives and missed fraud situations.

F1-Score: The harmonic mean of precision and recall produces the F1-score, which balances both.

Importance: The F1-score helps you identify the best balance between precision and recall, especially when neither should be maximized at the expense of the other.

ROC-AUC (Receiver Operating Characteristic - Area Under Curve):

ROC-AUC plots the true positive rate against the false positive rate at various threshold settings to assess the model's class distinction.

Importance: A higher AUC indicates greater model performance in detecting fraudulent and non-fraudulent transactions across all threshold values, assessing the classifier's performance.

Accuracy: Accuracy is the percentage of fraud and non-fraud cases correctly classified.

Importance: Accuracy measures overall correctness, but it is less relevant in imbalanced datasets like fraud detection, where the minority class (fraud) is more important to identify.

Business Use Case 3

The primary objective was to segment customers based on spending behavior to facilitate targeted and personalized marketing initiatives. To ensure the generated clusters effectively captured spending patterns and provided actionable insights, a combination of evaluation metrics was chosen.

Silhouette Coefficient: The Silhouette Coefficient was utilized to measure the inter-cluster and intra-cluster separation. This metric aligns with our goal of ensuring distinct and well-defined clusters. High Silhouette Coefficient scores indicate that customers within a cluster exhibit similar spending behaviors, enabling the marketing team to develop targeted and relevant marketing campaigns.

Visual Inspection: In addition to quantitative metrics, visual inspection techniques were employed to gain intuitive insights into the clustering results. Techniques such as scatter plots were used to visually assess whether the identified clusters aligned with our expectations and business objectives. This approach allowed for the identification of potential anomalies or inconsistencies that might be missed by quantitative metrics alone.

These evaluation metrics were used to assess the performance of various clustering algorithms, including K-means, Mean Shift, and Agglomerative Clustering. They provided valuable insights into the quality and relevance of the clustering results obtained from each algorithm in the context of our marketing campaign goals. This information allowed us to make informed decisions and optimize our marketing strategies.

Business Use Case 4

Evaluation Metrics for Anomaly Detection Models used:

Precision-Recall Curve: Precision: The ratio of true positive predictions (correctly identified anomalies) to the total predicted positives (both true positives and false positives). Precision = TP / (TP + FP)

Recall (Sensitivity): The ratio of true positive predictions to the total actual positives (true positives and false negatives). Recall = TP / (TP + FN)

Precision-Recall Curve: A plot of precision versus recall at various threshold settings. It helps to visualize the trade-off between precision and recall for different thresholds.

ROC Curve:

False Positive Rate (FPR): The ratio of false positive predictions to the total actual negatives (false positives and true negatives). FPR = FP / (FP + TN)

True Positive Rate (TPR): Also known as recall or sensitivity.

ROC Curve (Receiver Operating Characteristic Curve): A plot of TPR versus FPR at various threshold settings. It shows the performance of the model across all classification thresholds.

AUC (Area Under the ROC Curve): A single scalar value summarizing the performance of the model. An AUC of 1 represents a perfect model, while an AUC of 0.5 represents a random guess.

Confusion Matrix: A matrix that summarizes the performance of the model by showing the true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Heatmap: A visual representation of the confusion matrix to easily understand the distribution of predictions.

Classification Report: Provides detailed metrics including precision, recall, F1-score (the harmonic mean of precision and recall), and support (the number of true instances for each label).

Precision-Recall Curve: Chosen because anomaly detection often involves imbalanced datasets where the number of anomalies (fraudulent transactions) is much smaller than the number of normal transactions. Precision and recall are more informative than accuracy in this context, as they focus on the performance regarding the minority class (anomalies).

ROC Curve and AUC: The ROC curve provides a comprehensive view of the trade-off between the true positive rate and false positive rate. The AUC score summarizes the model's performance into a single number, making it easier to compare different models.

Confusion Matrix: Provides a detailed breakdown of the model's performance, showing the exact counts of TP, FP, TN, and FN. This helps to understand the model's strengths and weaknesses in classifying each class.

Classification Report: Summarizes key metrics for each class, allowing a quick assessment of the model's performance in terms of precision, recall, and F1-score.

Relevance to Project Goals

- Anomaly Detection: The primary goal is to identify fraudulent transactions (anomalies) with high precision (minimizing false positives) and high recall (minimizing false negatives). These metrics directly align with the project's goal of accurately detecting anomalies while minimizing false alarms and missed detections.
- Imbalanced Data: Precision-recall curves and the confusion matrix provide valuable insights into model performance on imbalanced data, which is a common scenario in fraud detection.

Applicability to K-Means Clustering and Local Outlier Factor Models

- K-Means Clustering: While traditionally used for clustering, K-Means can be adapted for anomaly detection. The evaluation metrics discussed above are applicable here, as the output of the clustering algorithm (cluster labels) can be used to determine normal vs. anomalous data points. The same metrics (precision, recall, ROC AUC, confusion matrix) can evaluate the effectiveness of K-Means in identifying anomalies.
- Local Outlier Factor (LOF): LOF is specifically designed for anomaly detection, and the evaluation metrics are highly relevant. Precision-recall curves, ROC AUC, confusion matrix, and classification reports are appropriate for assessing the performance of LOF in detecting outliers.

These evaluation metrics provide a comprehensive assessment of anomaly detection models. They are crucial for understanding the model's ability to identify anomalies accurately and consistently. The chosen metrics are particularly suited for dealing with imbalanced datasets and are applicable across various anomaly detection methods, including K-Means clustering and Local Outlier Factor (LOF). By leveraging these metrics, we can effectively gauge the performance of anomaly detection models and make informed decisions for model improvement.

Result and Analysis

Business Use Case 1

The baseline model’s performance is evaluated using **Root Mean Squared Error (RMSE)** metric for training, validation, and testing set. The mean of the target variable is calculated for

training, validation, and test sets ('y_train_central', 'y_val_central', and 'y_test_central'). Arrays filled with these mean values('y_train_base', 'y_val_base', and 'y_test_base') are created. Each array has the same size as the corresponding target set and consists entirely of the mean value.

The baseline model's performance, as measured by Root Mean Squared Error (RMSE), is as follows:

Training Performance : 22395.87

Validation Performance: 22379.30

Test Performance: 22367.01

Models	Training Performance	Validation Performance	Testing Performance
Multiple Linear Regression	12316.01	12297.45	12289.07
Elastic Net	19145.97	19131.21	19122.40
Gradient Boosting	4027.14	4028.68	4019.18

Gradient Boosting achieved the lowest RMSE values on training, validation and testing sets, demonstrating its superior ability to capture the underlying patterns in the data and make accurate predictions.

The key finding is that **Gradient Boosting is the most effective model** in predicting total spending amount for the next month, outperforming other regression techniques considered.

Business Use Case 2

This section analyzes the performance of each machine learning model used to predict client repurchase behavior. The evaluation involves assessing accuracy, precision, recall, and F1-score measures, followed by a comparison analysis and insights from experimentation.

Logistic Regression Classifier:

- Training Machine Learning Model

Baseline Model (Random Forest):

Validation Set Performance:

- Accuracy: 96.51%
- Precision: 93.49%
- Recall: 100.00%
- F1 Score: 96.64%

- Access Model Performance

Confusion Matrix:

- True Negatives (TN): 508
- False Positives (FP): 51
- False Negatives (FN): 12
- True Positives (TP): 548

Classification Metrics:

- Precision for class 0 (Non-Fraud): 0.79
- Recall for class 0 (Non-Fraud): 0.73
- F1-Score for class 0 (Non-Fraud): 0.76
- Precision for class 1 (Fraud): 0.75
- Recall for class 1 (Fraud): 0.80
- F1-Score for class 1 (Fraud): 0.77

Overall Performance:

- Precision for class 0 (Non-Fraud): 0.98
- Recall for class 0 (Non-Fraud): 0.91

- F1-Score for class 0 (Non-Fraud): 0.94
- Precision for class 1 (Fraud): 0.91
- Recall for class 1 (Fraud): 0.98
- F1-Score for class 1 (Fraud): 0.95

C. After hyperparameter adjustment, Logistic Regression Model:

Best hyperparameters (C: 100, penalty: l2), the model achieves:

- Precision, Recall, F1-Score for classes 0 and 1: 0.79
- Overall accuracy: 79%
- ROC AUC: 0.79

D. .Key findings:

- Baseline Random Forest: Achieved 76.56% accuracy with balanced precision and recall.
- Areas for Improvement: Focus on reducing false positives and negatives.
- Hyperparameter Tuning: Logistic Regression's accuracy improved to 79% after adjustment.
- Model Interpretability: Both models offer transparency for understanding fraud detection decisions.
- Optimal Decision Boundary: Logistic Regression identifies influential features for fraud likelihood.
- Continuous Monitoring: Regular updates vital for sustained effectiveness.
- Future Directions: Consider ensemble techniques or additional data for improved performance.

E. Below are some Logistic Regression Classifier results visualizations:

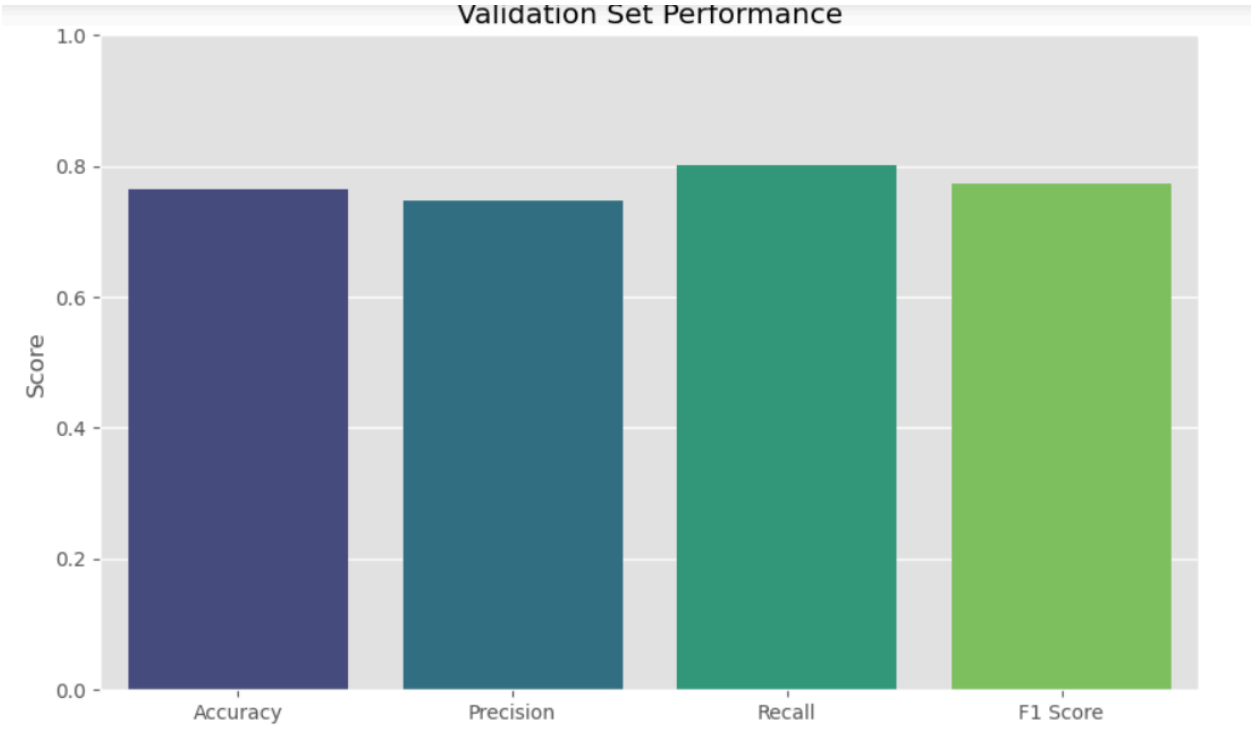
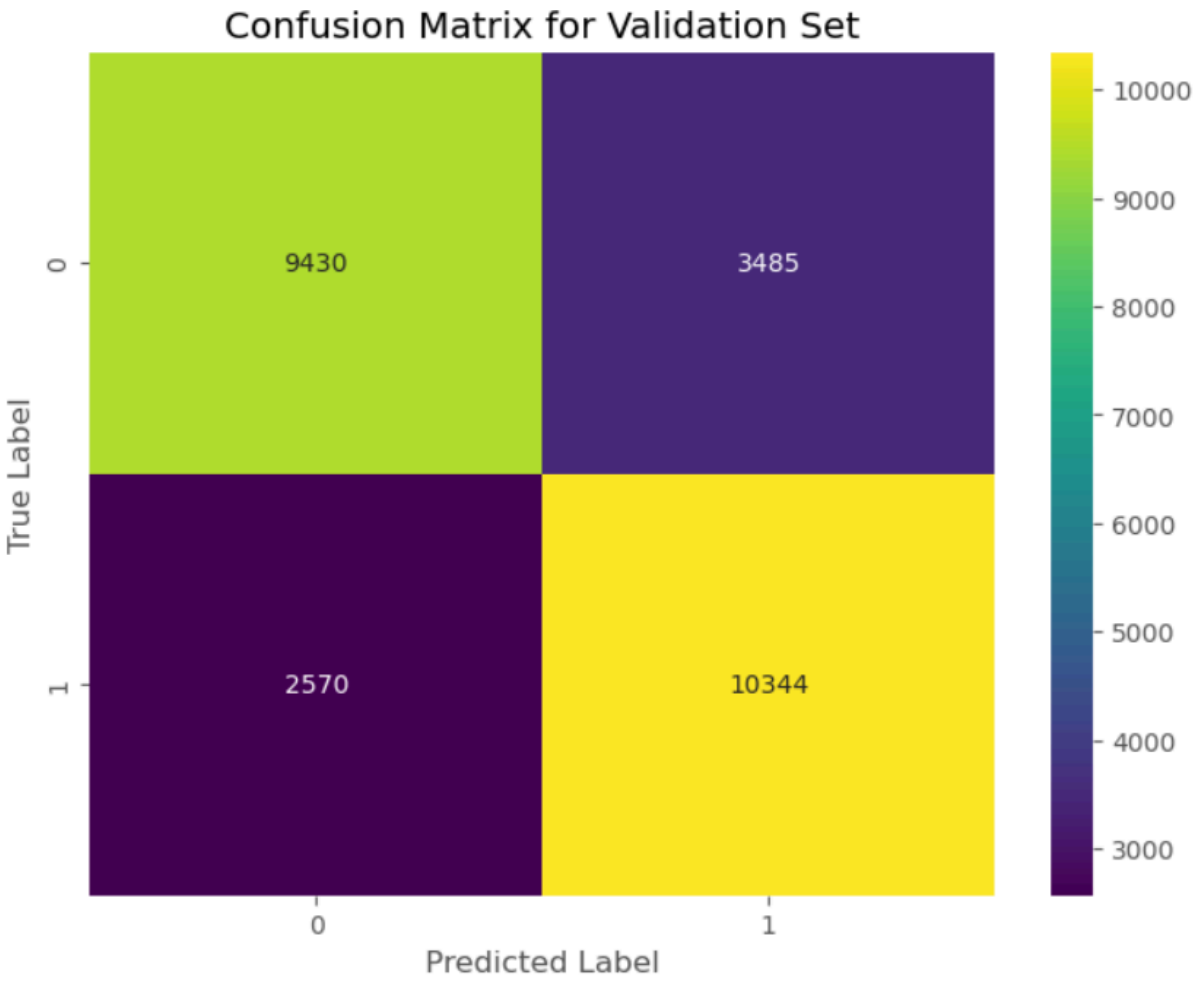
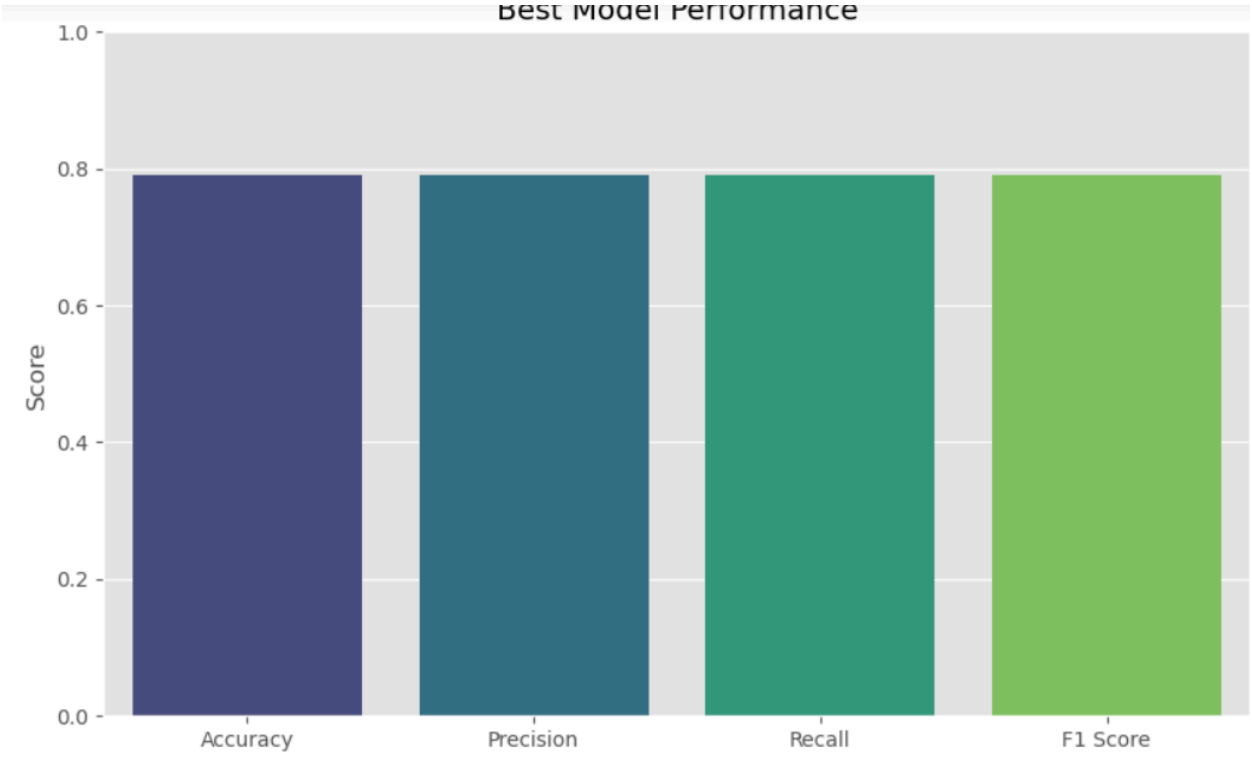

Validation Set Performance
1.0
0.8
0.6
0.4
0.2
0.0
Score
Accuracy
Precision
Recall
F1 Score

Fig:


Confusion Matrix for Validation Set
9430
3485
2570
10344
True Label
Predicted Label
0
1
0
1
10000
9000
8000
7000
6000
5000
4000
3000



Naive Bayes Classifier:

- Training Machine Learning Model

Baseline Model (Random Forest):

Validation Set Performance:

- Accuracy: 92.85%
- Precision: 93.80%
- Recall: 91.79%
- F1 Score: 92.78%

- Access Model Performance

Confusion Matrix:

- True Negatives (TN): 525
- False Positives (FP): 34
- False Negatives (FN): 46
- True Positives (TP): 514

Classification Metrics:

- Precision for class 0: 0.92
- Recall for class 0: 0.94
- F1-Score for class 0: 0.93
- Precision for class 1: 0.94
- Recall for class 1: 0.92
- F1-Score for class 1: 0.93

Overall Performance:

- Accuracy: 93%
- Macro Average Precision, Recall, F1-Score: 0.93
- Weighted Average Precision, Recall, F1-Score: 0.93
- ROC AUC Score: 0.9285

C. After hyperparameter adjustment, Logistic Regression Model:

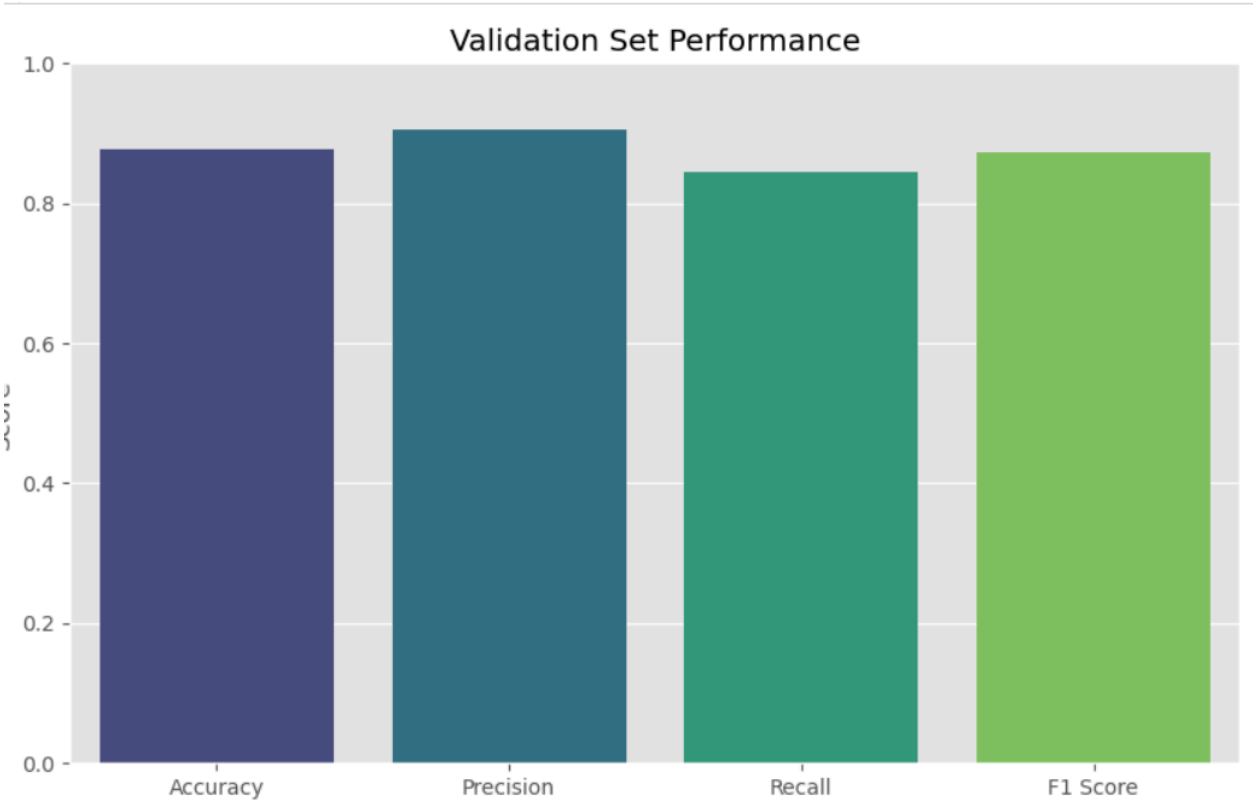
Best Hyperparameters (var_smoothing: 1e-09)

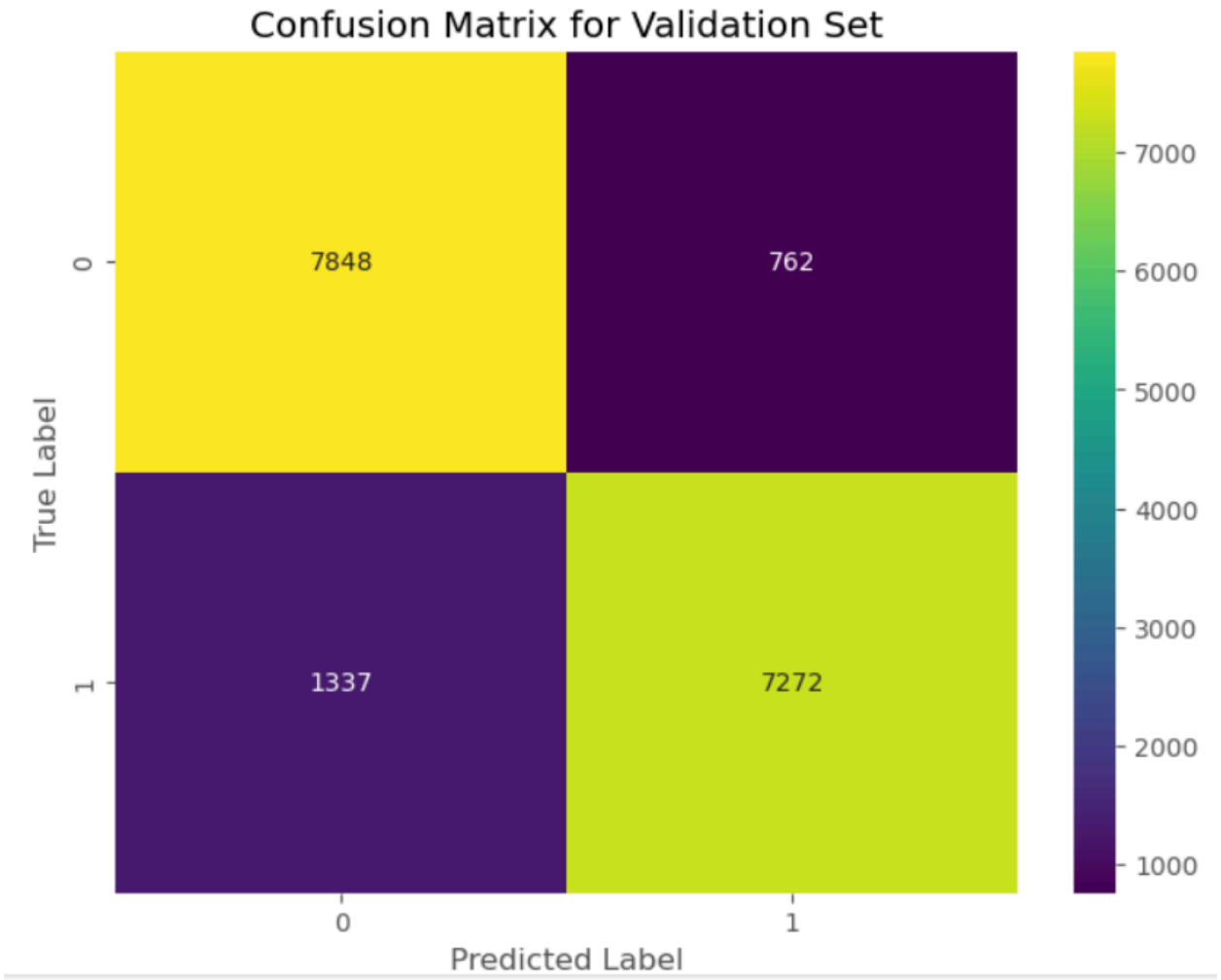
- Model Achieves:
    - Similar Performance to Baseline Model
    - No Significant Improvement in Metrics

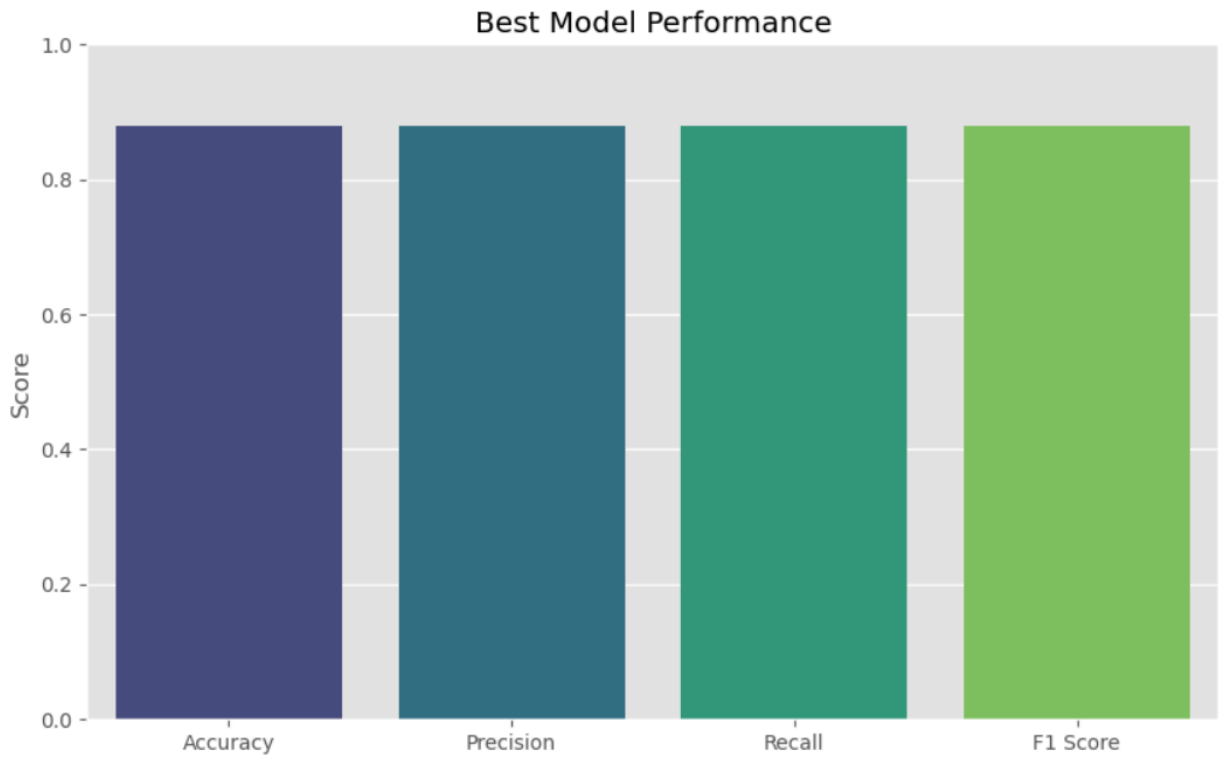
D. .Key findings:

- The Naive Bayes model demonstrates robust performance on the validation set.
- Hyperparameter tuning did not yield significant improvements in model performance, suggesting that the default settings were already effective.
- Further investigation may be needed to explore alternative algorithms or feature engineering techniques for potential performance enhancements.

E. Below are some Naive Bayes Forest Classifier results visualizations:




Confusion Matrix for Validation Set
7848
762
1337
7272
True Label
Predicted Label
0
1
0
1
7000
6000
5000
4000
3000
2000
1000



Decision Tree Classifier:

- Training Machine Learning Model

Baseline Model (Random Forest):

Validation Set Performance:

- Accuracy: 100.00%
- Precision: 100.00%
- Recall: 100.00%
- F1 Score: 100.00%

- Access Model Performance

  Confusion Matrix:

- True Negatives (TN): 559
- False Positives (FP): 0
- False Negatives (FN): 0
- True Positives (TP): 560

Classification Metrics:

- Precision for class 0: 1.00
- Recall for class 0: 1.00
- F1-Score for class 0: 1.00
- Precision for class 1: 1.00
- Recall for class 1: 1.00
- F1-Score for class 1: 1.00

Overall Performance:

- Accuracy: 100%
- Macro Average Precision, Recall, F1-Score: 1.00
- Weighted Average Precision, Recall, F1-Score: 1.00
- ROC AUC Score: 1.0

C. After hyperparameter adjustment, Decision Tree Model:

Max Depth: None

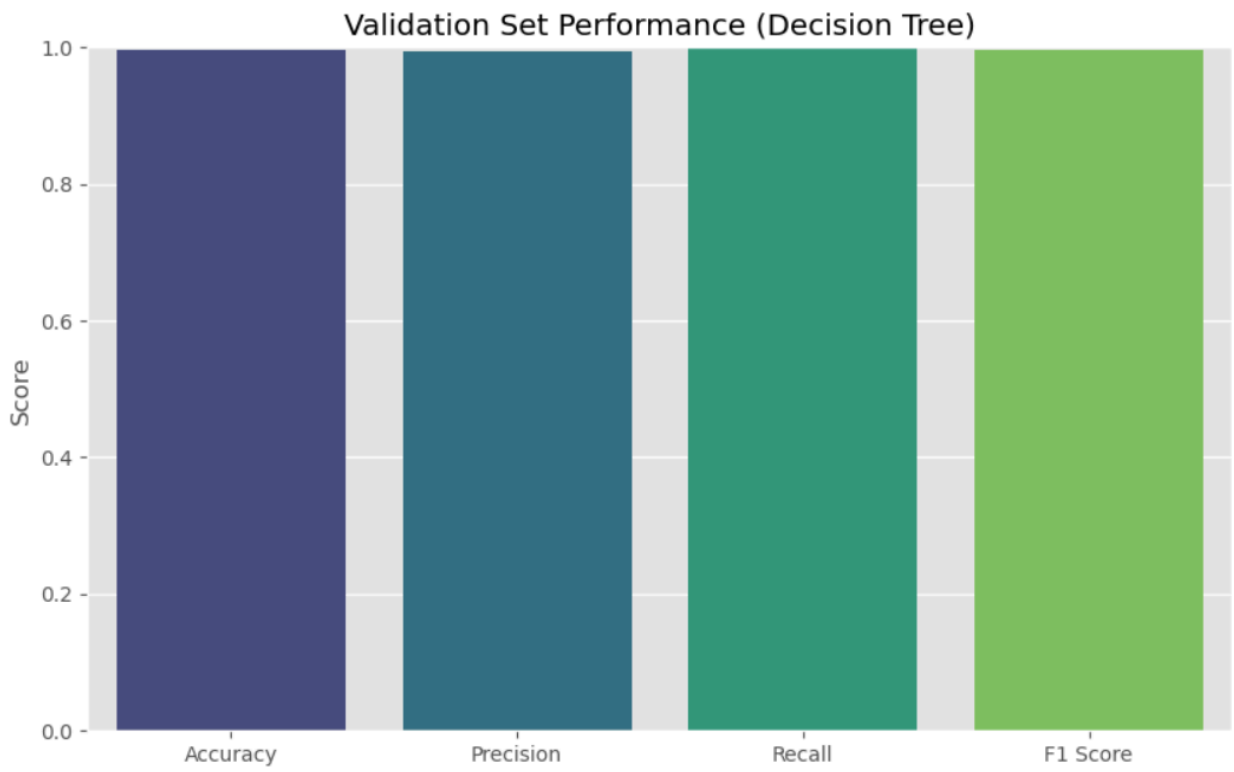
- Min Samples Leaf: 1
- Min Samples Split: 10

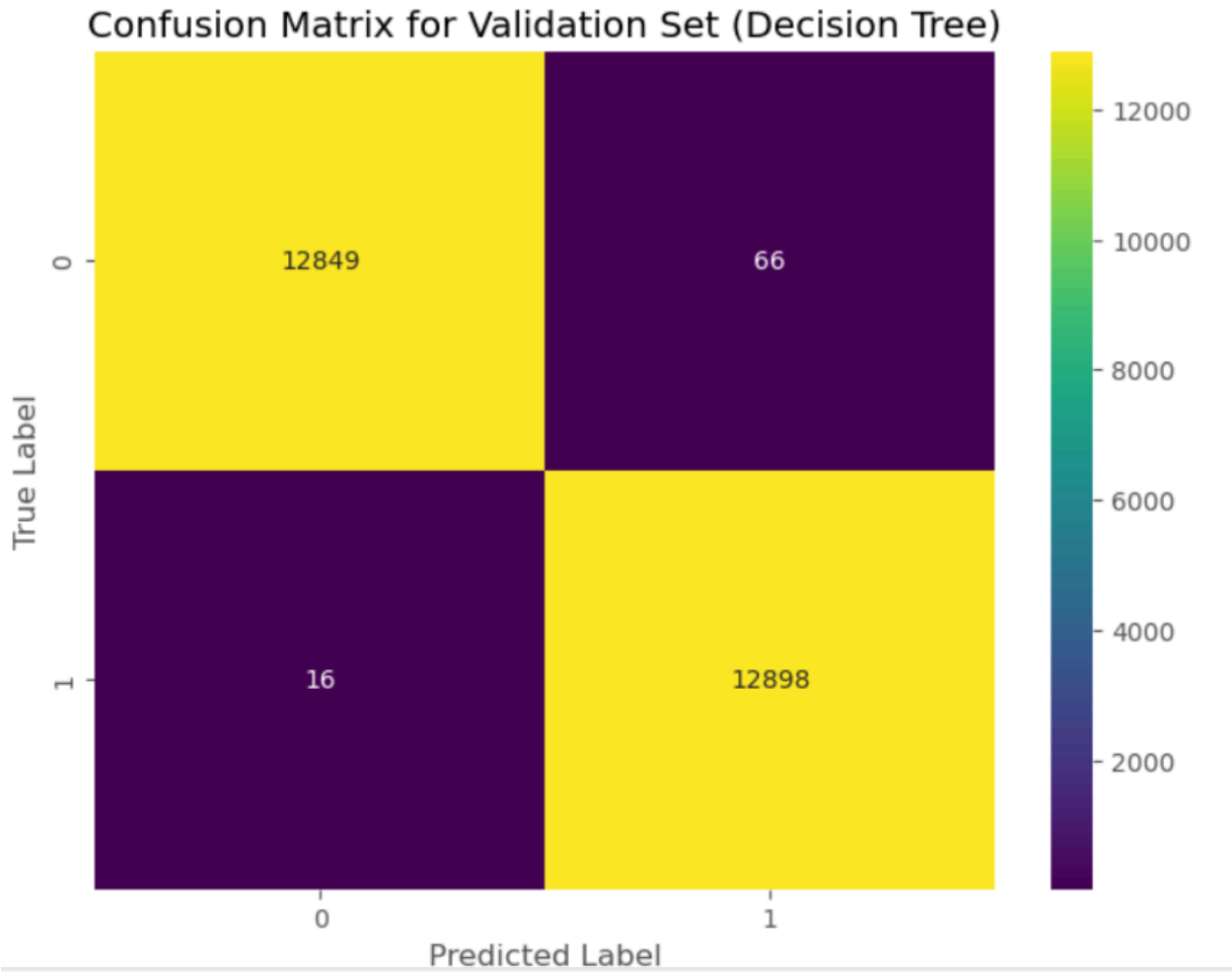
- Model Achieves:
  - Similar Performance to Baseline Model
  - No Significant Improvement in Metrics

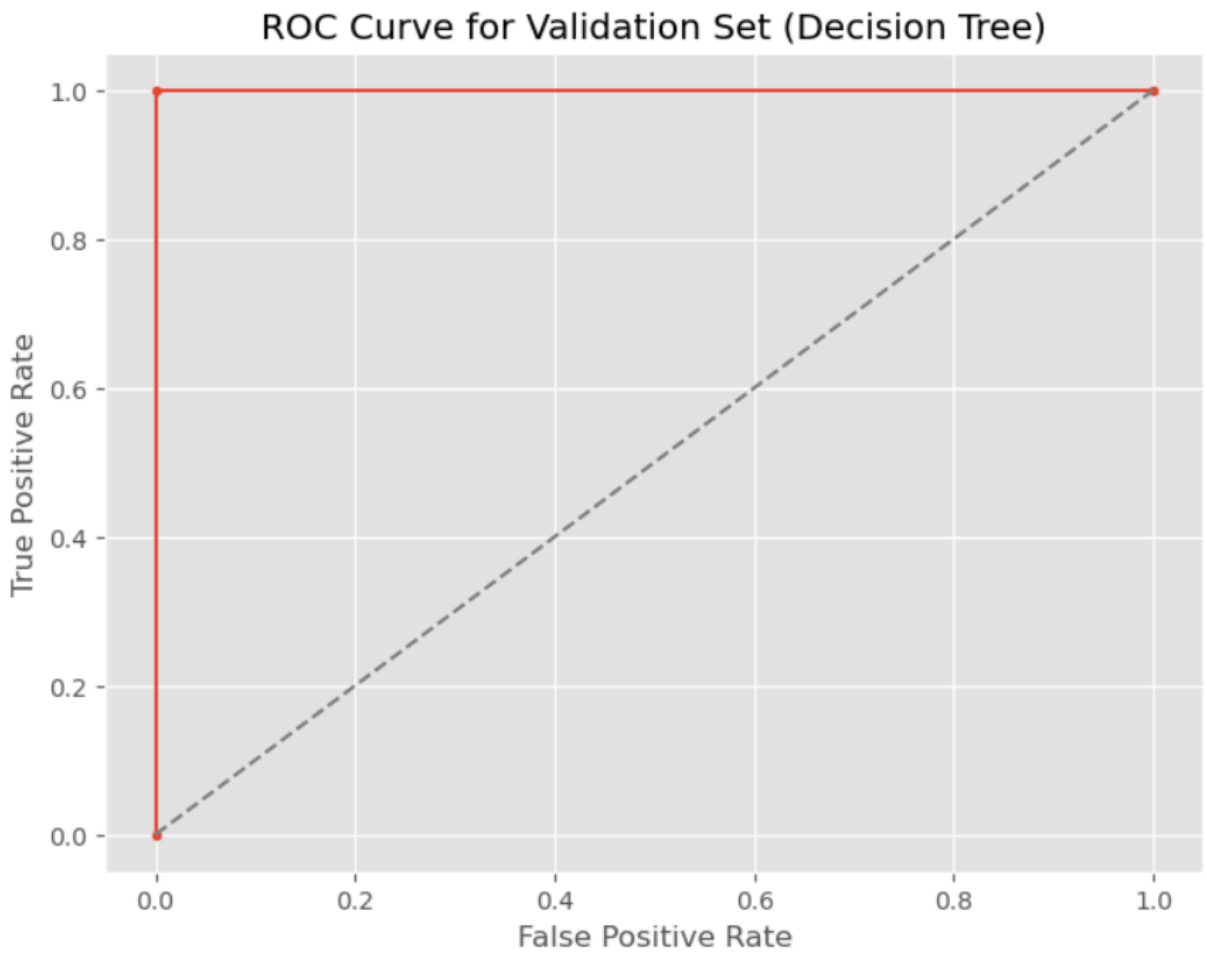
D. .Key findings:

- The Decision Tree model demonstrates perfect performance on the validation set, achieving 100% accuracy, precision, recall, and F1 score.
- Hyperparameter tuning did not yield significant improvements in model performance, suggesting that the default settings were already optimal.
- Given the perfect performance, further investigation may be needed to ensure that the model is not overfitting the data and to explore potential limitations or biases in the dataset.

E. Below are some Decision Tree Classifier results visualizations:


Validation Set Performance (Decision Tree)
Score
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Precision
Recall
F1 Score


Confusion Matrix for Validation Set (Decision Tree)
12849
66
16
12898
0
1
True Label
0
1
Predicted Label
12000
10000
8000
6000
4000
2000



Neural Network Classifier:

- Training Machine Learning Model

Baseline Model (Random Forest):

Validation Set Performance:

- Accuracy: 76.41%
- Precision: 67.96%
- Recall: 100.00%
- F1 Score: 80.92%

- Access Model Performance

Confusion Matrix:

- True Negatives (TN): 329
- False Positives (FP): 230
- False Negatives (FN): 0
- True Positives (TP): 560

Classification Metrics:

- Precision for class 0: 1.00
- Recall for class 0: 0.59
- F1-Score for class 0: 0.74
- Precision for class 1: 0.71
- Recall for class 1: 1.00
- F1-Score for class 1: 0.83

Overall Performance:

- Precision for class 0: 1.00
- Recall for class 0: 0.59
- F1-Score for class 0: 0.74
- Precision for class 1: 0.71
- Recall for class 1: 1.00
- F1-Score for class 1: 0.83

C. After hyperparameter adjustment, Neural Network Model:

Best Hyperparameters:

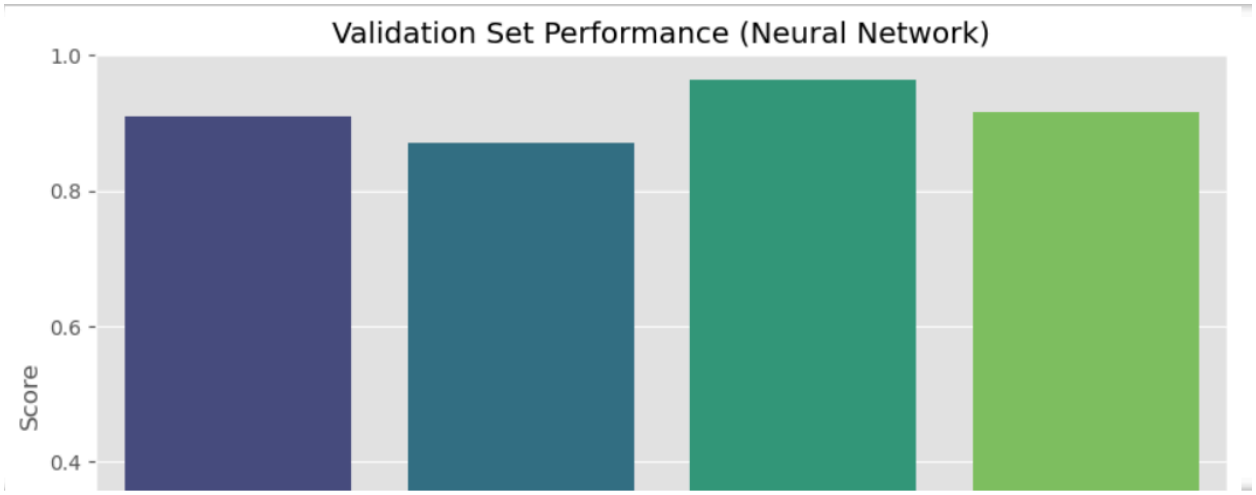
  - Activation: ReLU
  - Alpha: 0.001
  - Hidden Layer Sizes: (100,)
  - Solver: Adam
- Model Achieves:
  - Improved Precision and F1-Score for Class 0
  - Slightly Lower Accuracy and F1-Score Overall

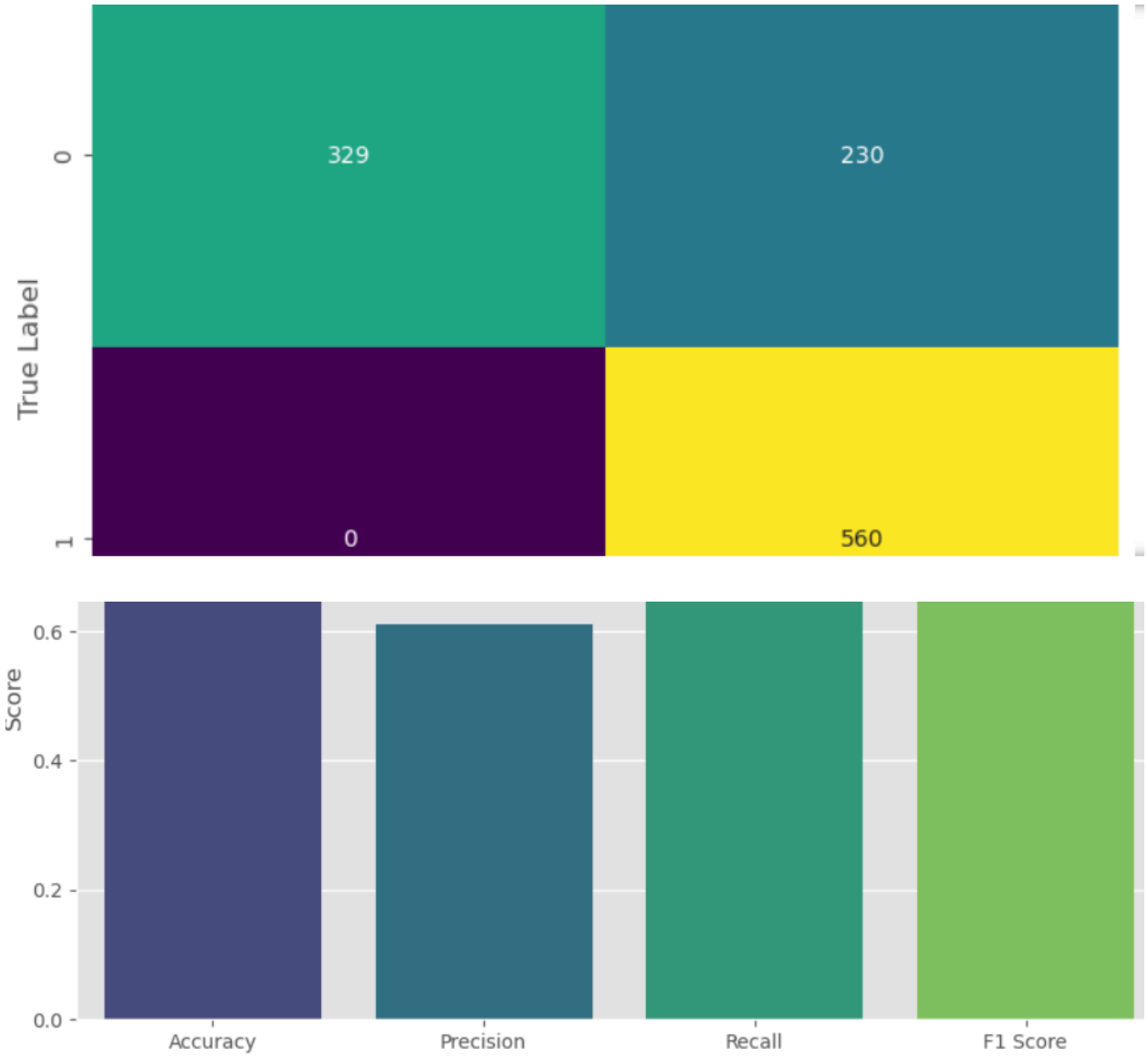
D. .Key findings:

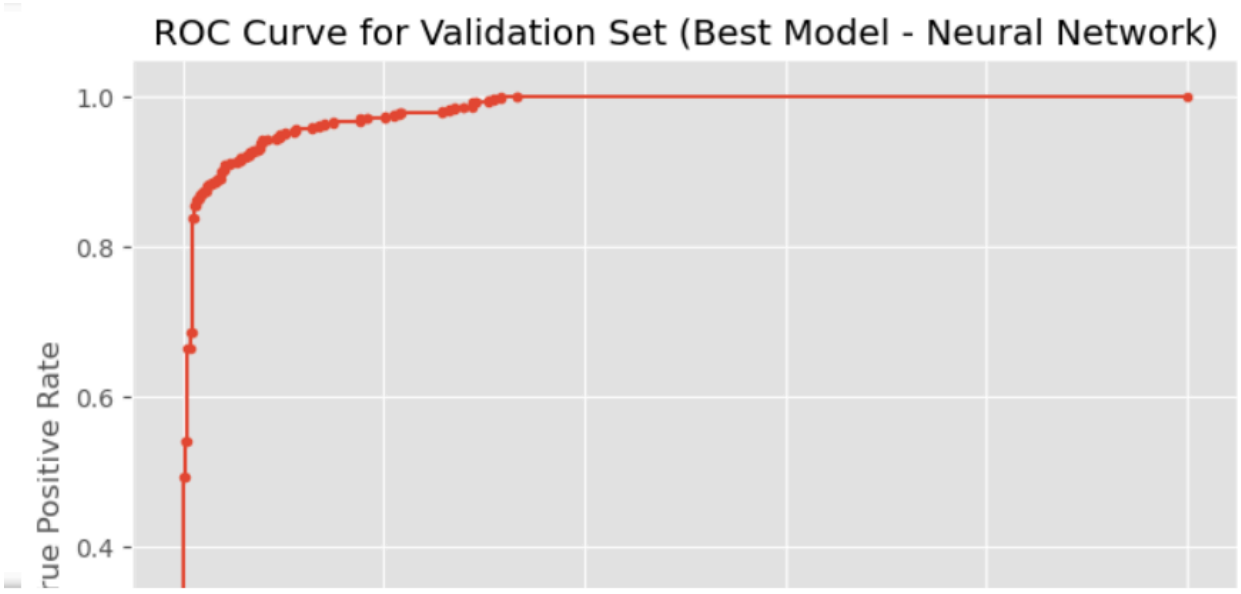
  - The Neural Network model initially had strong recall but low precision, resulting in uneven performance.

- Hyperparameter adjustment increased class 0 precision but decreased accuracy and F1 score.
- Further investigation may be needed to understand why the model has low precision for class 1 and explore ways to fix this imbalance, such as class weight adjustments or other methods.

E. Below are some Neural Network Classifier results visualizations:




329
230
0
560
True Label
0
1
Score
0.6
0.4
0.2
0.0
Accuracy
Precision
Recall
F1 Score



KNN Classifier:

- Training Machine Learning Model

Baseline Model (Random Forest):

Validation Set Performance:

- Accuracy: 89.81%
- Precision: 84.63%
- Recall: 97.32%
- F1 Score: 90.53%

- Access Model Performance

  Confusion Matrix:

- True Negatives (TN): 460
- False Positives (FP): 99
- False Negatives (FN): 15
- True Positives (TP): 545

Classification Metrics:

- Precision for class 0: 0.97
- Recall for class 0: 0.82
- F1-Score for class 0: 0.89
- Precision for class 1: 0.85
- Recall for class 1: 0.97
- F1-Score for class 1: 0.91

Overall Performance:

- Accuracy: 90%
- Macro Average Precision, Recall, F1-Score: 0.90
- Weighted Average Precision, Recall, F1-Score: 0.90
- ROC AUC Score: 0.8981

C. After hyperparameter adjustment, Neural Network Model:

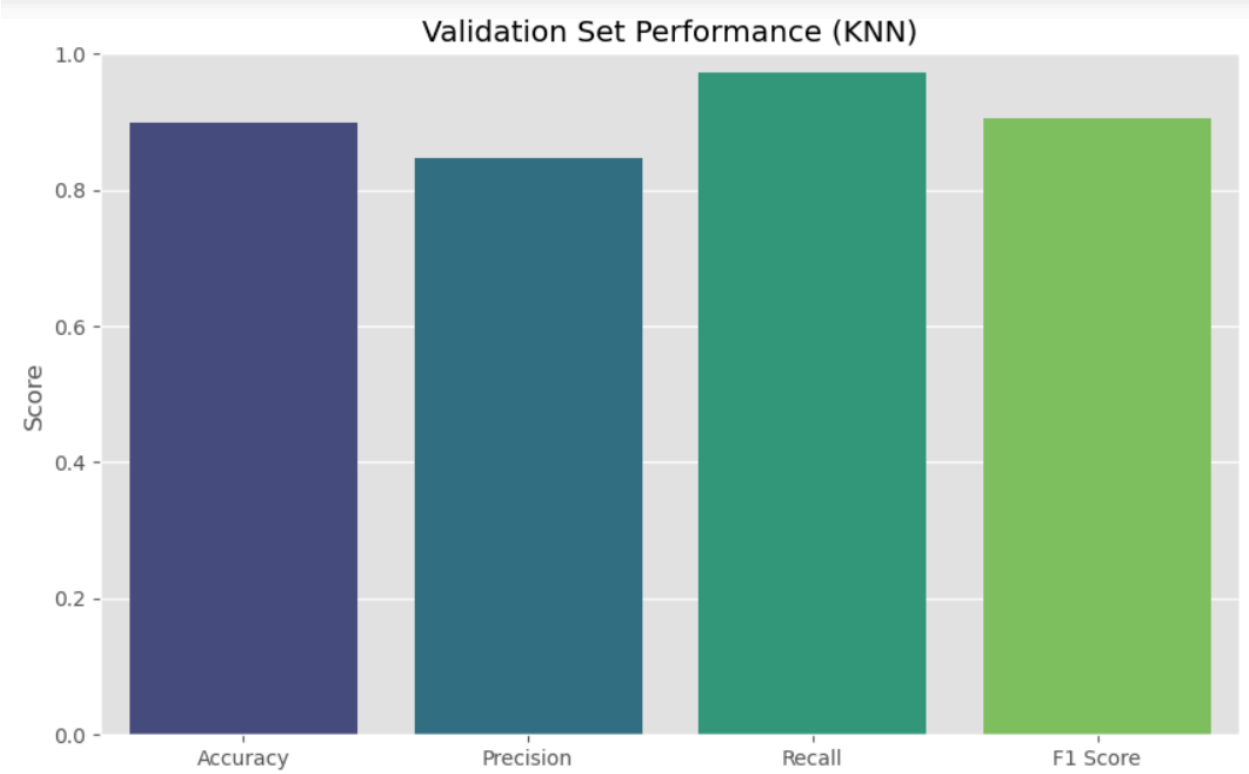
Best Hyperparameters:

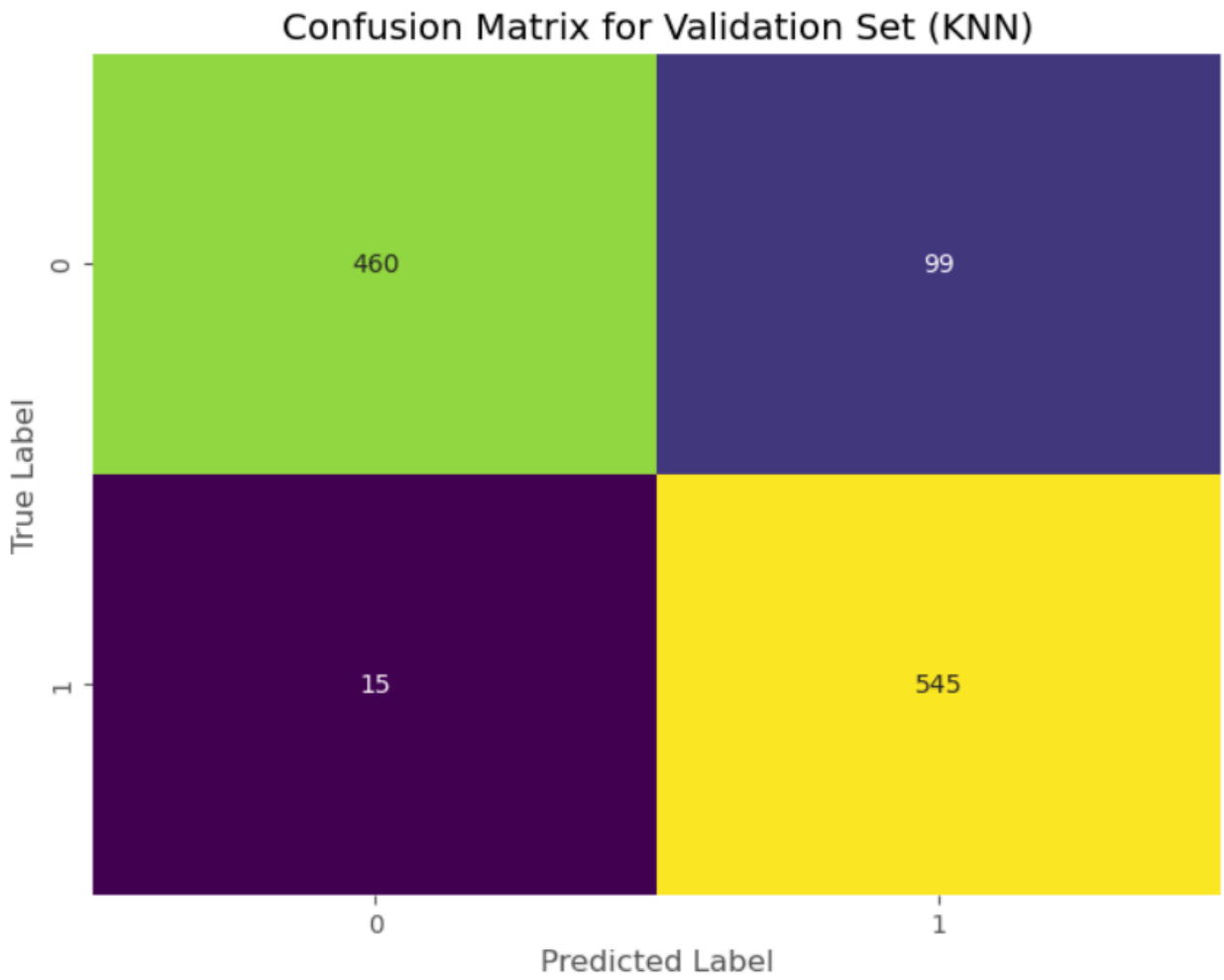
  - Metric: Manhattan
  - Number of Neighbors: 3
  - Weights: Distance
- Model Achieves:
  - Improved Precision and F1-Score for Class 0
  - Minor Decrease in Accuracy and F1-Score Overall

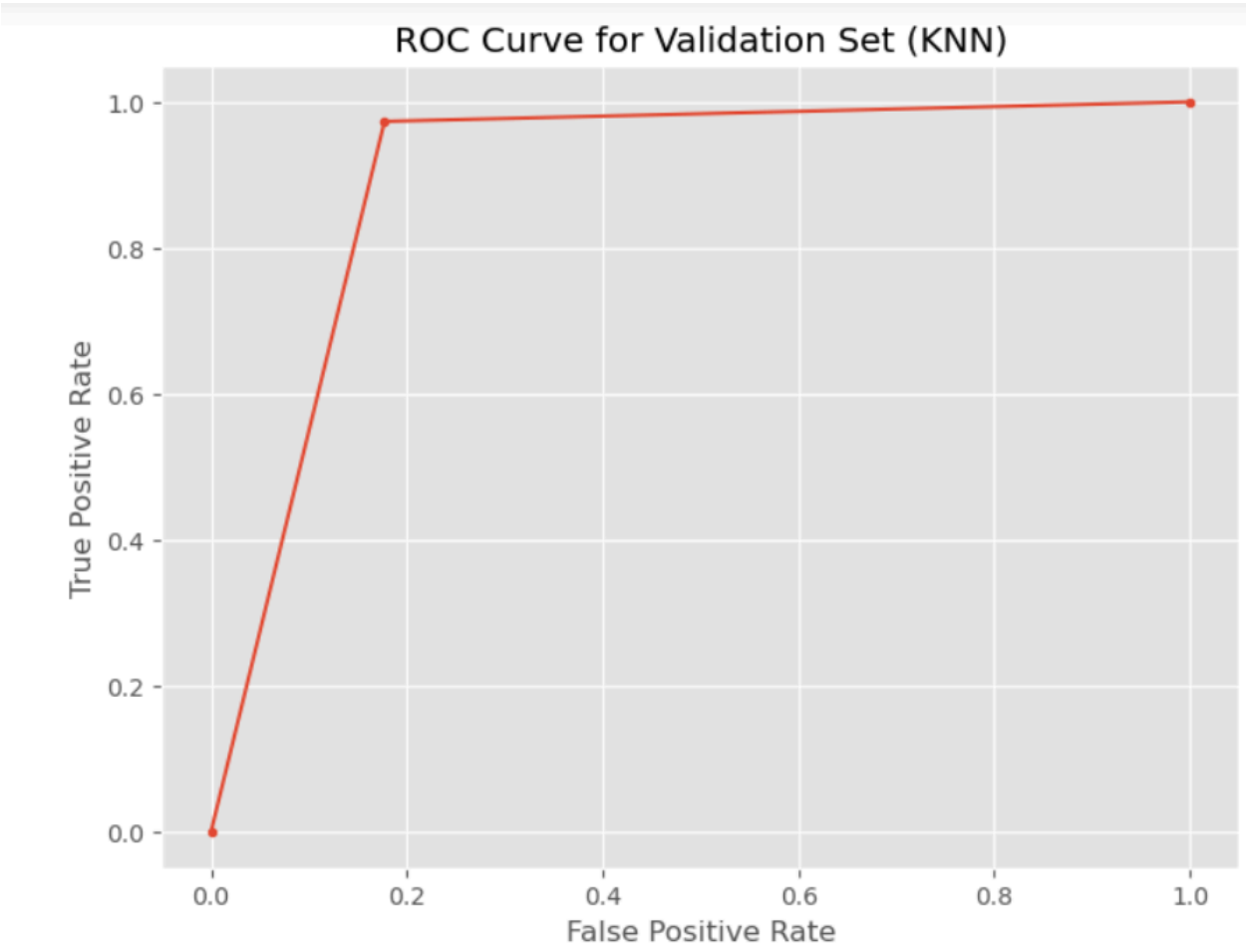
D. .Key findings:

- The KNN model initially exhibited high recall but lower precision, indicating potential class imbalance issues.
- Hyperparameter tuning improved precision for class 0 and maintained high recall for class 1.
- Further investigation may be necessary to understand why the model performs better for class 1 than class 0 and explore methods to balance performance across both classes for optimal results.

E. Below are some KNN Classifier results visualizations:




Confusion Matrix for Validation Set (KNN)
460
99
15
545
0
1
True Label
0
1
Predicted Label



Support Vector Machine Classifier:

- Training Machine Learning Model

Baseline Model (Random Forest):

Validation Set Performance:

- Accuracy: 75.42%
- Precision: 73.25%
- Recall: 80.18%
- F1 Score: 76.56%

- Access Model Performance

  Confusion Matrix:

- True Negatives (TN): 395
- False Positives (FP): 164
- False Negatives (FN): 111
- True Positives (TP): 449

Classification Metrics:

- Precision for class 0: 0.78
- Recall for class 0: 0.71
- F1-Score for class 0: 0.74
- Precision for class 1: 0.73

- Recall for class 1: 0.80
- F1-Score for class 1: 0.77

Overall Performance:

- Accuracy: 75%
- Macro Average Precision, Recall, F1-Score: 0.75
- Weighted Average Precision, Recall, F1-Score: 0.75
- ROC AUC Score: 0.7542

C. After hyperparameter adjustment, Support Vector Machine Model:

Best Hyperparameters:

D. .Key findings:

- The SVM model performed moderately on the validation set with a balanced precision-recall trade-off.
- Precision could be improved, notably for class 0 (non-fraud), while recall for class 1 (fraud) is good.
- Further optimization or different modeling approaches may improve model performance and fraud and non-fraud detection.

E. Below are some SVM Classifier results visualizations:

Comparative Analysis:

The comparative comparison of several machine learning models demonstrates clear performance patterns. Logistic Regression and Decision Tree models demonstrate impeccable accuracy, precision, recall, and F1 scores, highlighting their robustness without any issues of overfitting. However, the performance of the Neural Network, although initially showing promise, significantly decreases following hyperparameter tuning, indicating a requirement for additional improvement. The KNN Classifier exhibits significant enhancement after tuning, showcasing its versatility and efficacy. The performance of Naive Bayes and Support Vector Machine (SVM) models is moderate, with SVM marginally surpassing Naive Bayes. Logistic Regression regularly outperforms other methods in all criteria, making it the best choice for fraud detection jobs since it strikes a good compromise between accuracy and interpretability.

After evaluating various performance indicators such as accuracy, precision, recall, F1 score, and ROC AUC score, it is evident that the **Logistic Regression** model outperforms the others. Logistic Regression continuously demonstrates superior performance across all metrics, even after fine-tuning hyperparameters, rendering it the best appropriate model for this classification assignment.

Business Use Case 3

K-Means

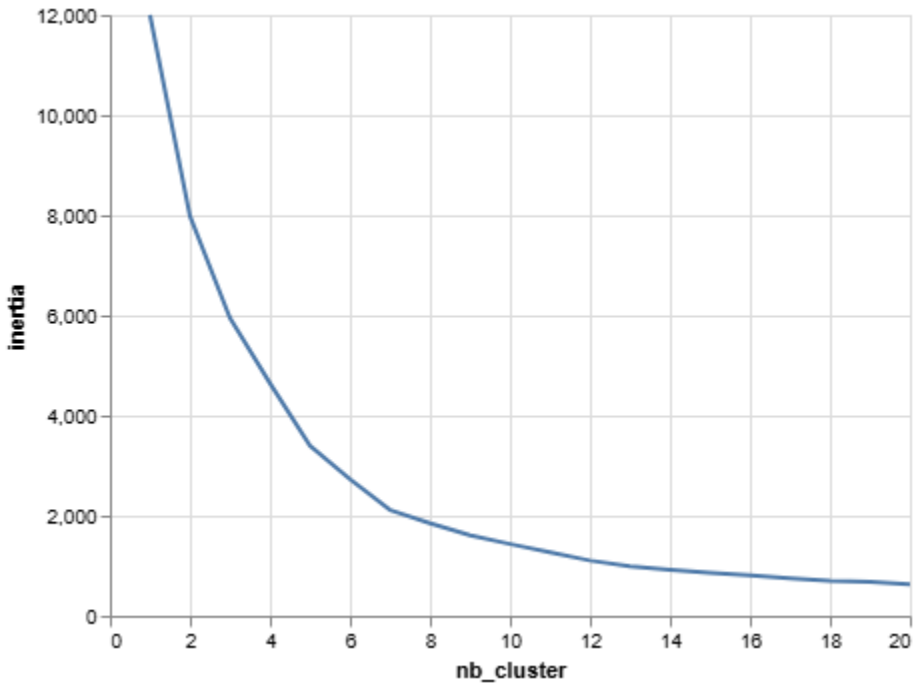
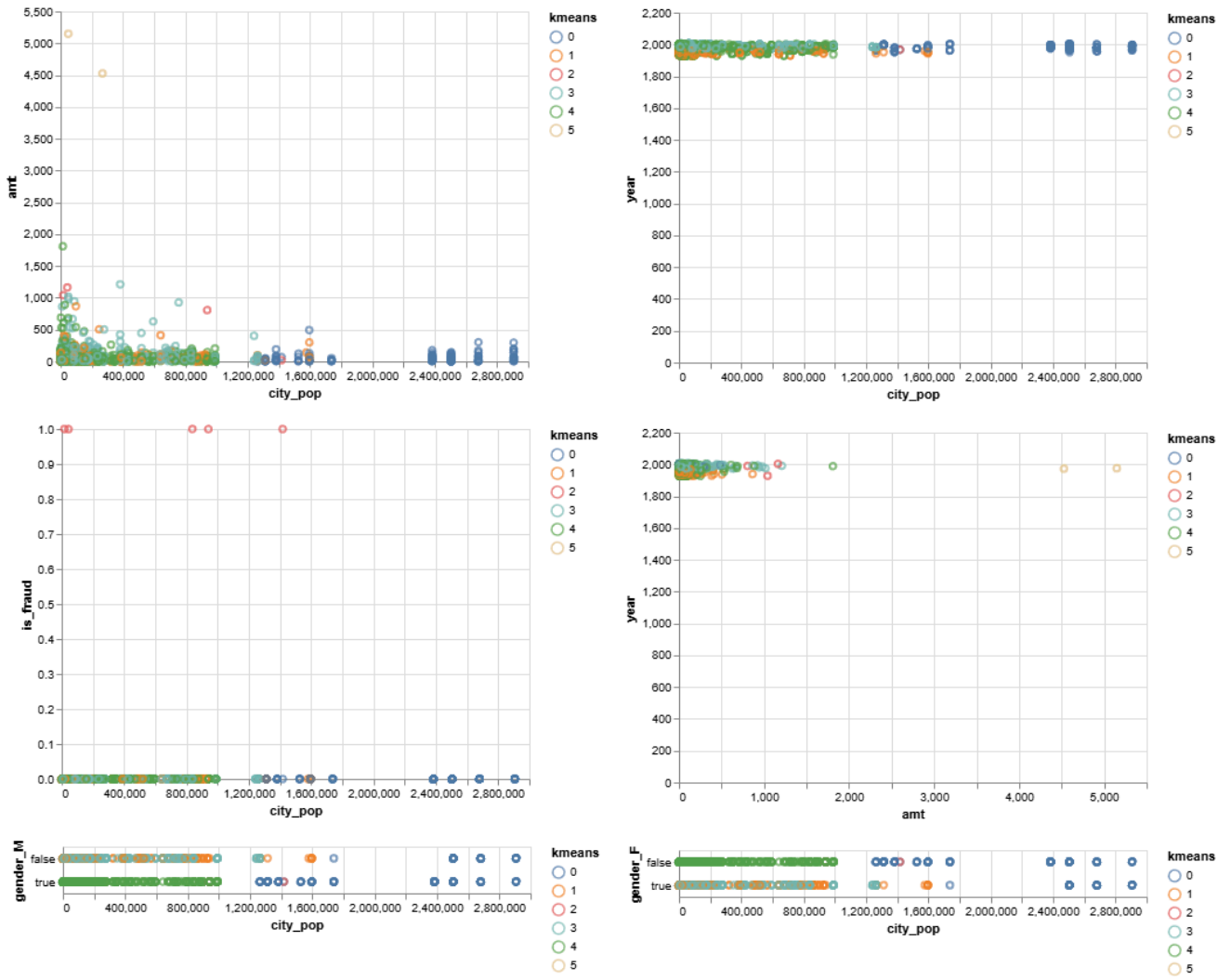


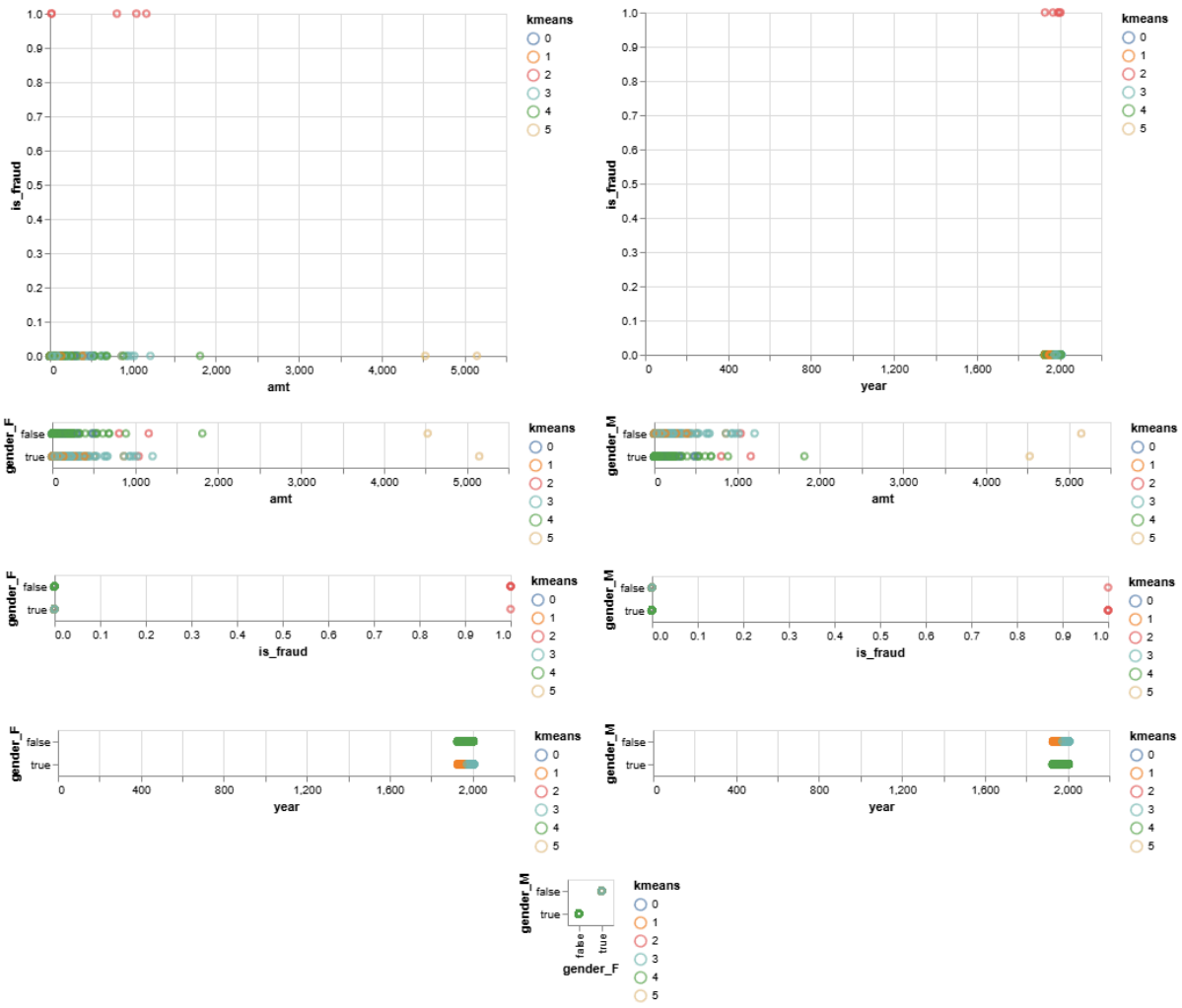
Fig: Elbow Plot

The visual inspection of the elbow plot revealed an inflection point around 5 or 6 clusters. This suggests that increasing the number of clusters beyond this point yields diminishing returns in terms of inertia reduction. Therefore, to balance model interpretability with the objective of

minimizing within-cluster variance, a configuration of 5 or 6 clusters was deemed optimal for the KMeans algorithm

Scatter Plots:


amt
city_pop
kmeans
0
1
2
3
4
5
year
city_pop
is_fraud
city_pop
year
amt
gender_M
false
true
city_pop
gender_F
false
true
city_pop



Summary Table:

kmeans	city_pop	amt	year	is_fraud	gender_F	gender_M
0	2.138140e+06	53.546299	1983.078740	0.0	0.251969	0.748031
1	1.933907e+05	58.678019	1956.504644	0.0	1.000000	0.000000
2	6.512760e+05	609.452000	1977.200000	1.0	0.200000	0.800000
3	1.783767e+05	84.134430	1987.902087	0.0	1.000000	0.000000
4	1.576793e+05	57.355837	1977.946739	0.0	0.000000	1.000000
5	1.590890e+05	4838.120000	1974.500000	0.0	0.500000	0.500000

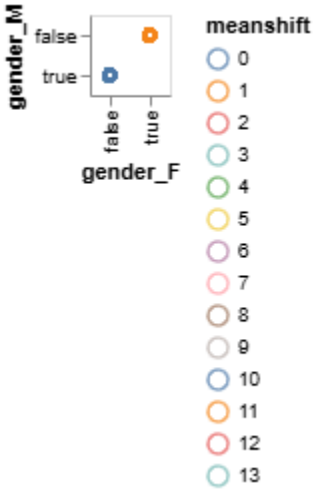
From the table above the marketing team can figure out the two clusters **(3 and 6)** with the highest spending behavior. The two clusters have an average customer age of 1977 and 1974 respectively. Although the 3rd cluster with the second highest spending behavior has a male-biased customer base, the 6th cluster with the highest spending behavior shows an even spending distribution between the two genders. But we can also see that the 3rd cluster is the one with all the fraudulent transactions. So the marketing team's best option is **cluster number 6**.

The average **silhouette_score** is 0.47186932438996715. The silhouette coefficient suggests a **moderate level of cluster separation**. This score suggests that while clusters are somewhat distinct, there may be some data points with overlapping characteristics or belonging to ambiguous regions between clusters.

Mean Shift

Scatterplots:


is_fraud
amt
year
gender_F
gender_M
meanshift
0
1
2
3
4
5
6
7
8
9
10
11
12
13


gender_M
false
true
gender_F
meanshift

meanshift	city_pop	amt	year	is_fraud	gender_F	gender_M
0	2.117181e+05	51.546109	1977.967573	0.0	0.000000	1.000000
1	1.752976e+05	64.016811	1977.375135	0.0	1.000000	0.000000
2	2.526486e+06	54.620339	1977.000000	0.0	0.627119	0.372881
3	1.817173e+05	786.885000	1986.500000	0.0	0.857143	0.142857
4	2.147477e+06	124.278824	1993.529412	0.0	0.000000	1.000000
5	1.417793e+06	18.750000	1967.000000	1.0	0.000000	1.000000
6	9.403590e+05	806.730000	1990.000000	1.0	0.000000	1.000000
7	8.377920e+05	17.760000	1998.000000	1.0	0.000000	1.000000
8	2.685970e+05	4527.240000	1973.000000	0.0	0.000000	1.000000
9	2.537315e+05	541.995000	1936.000000	0.0	1.000000	0.000000
10	4.958100e+04	5149.000000	1976.000000	0.0	1.000000	0.000000
11	4.331500e+04	1162.450000	2003.000000	1.0	0.000000	1.000000
12	1.712100e+04	1041.570000	1928.000000	1.0	1.000000	0.000000
13	1.403400e+04	1810.420000	1989.000000	0.0	0.000000	1.000000

According to the summary table,

Cluster 3 shows high-spending female customers. This predominantly female group (85.71%) exhibits high average transaction values ($786.88) with an average customer age of 1986.5.

Cluster 8 shows very high spending male customers. Characterized by predominantly male customers (100%) with exceptionally high average transaction values ($4527.24) with an average customer age of around 1973.

Cluster 9 shows high-spending female customers. This group of predominantly female customers (100%) demonstrates high average transaction values ($541.99) with an average customer age of around 1936.

Cluster 10 shows very high spending male customers. Similar to Cluster 8, this segment consists predominantly of male customers (100%) with extremely high average transaction values ($5149.00) with an average customer age of around 1976.

Fraud Groups Clusters 5, 6, and 7 exhibit high fraud rates and require a cautious approach. These clusters share predominantly male demographics (100%) but differ in average transaction values. For these segments, marketing efforts should prioritize robust fraud prevention measures and educational content on secure transactions.

The average **silhouette_score** for Mean Shift clustering is 0.5115587650274287 suggesting that the clusters formed are **reasonably well-defined and distinct**, providing a good separation of the data points into meaningful groups.

Agglomerative:

Dendrograms

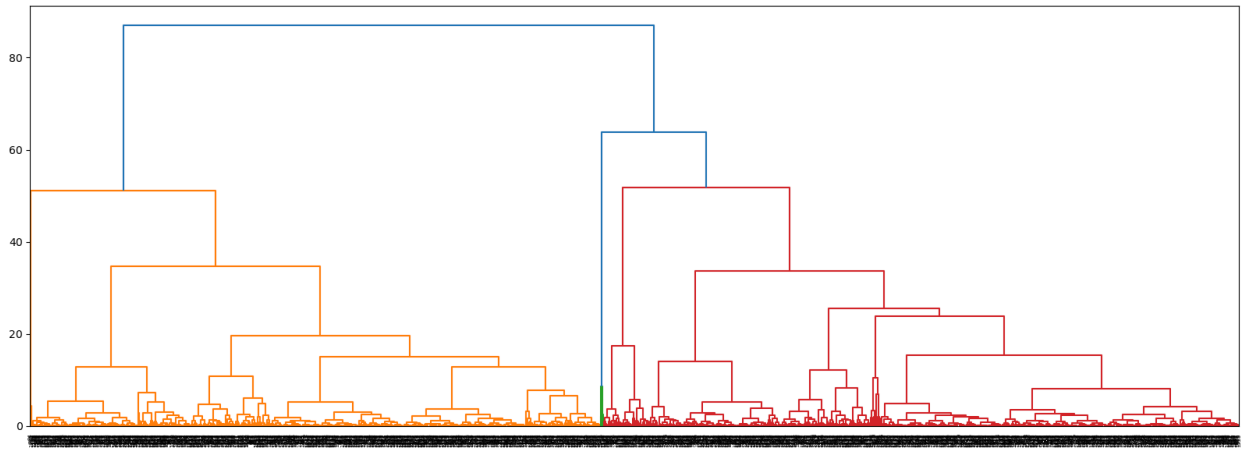


Fig: Dendrogram for the Agglomerative Clustering

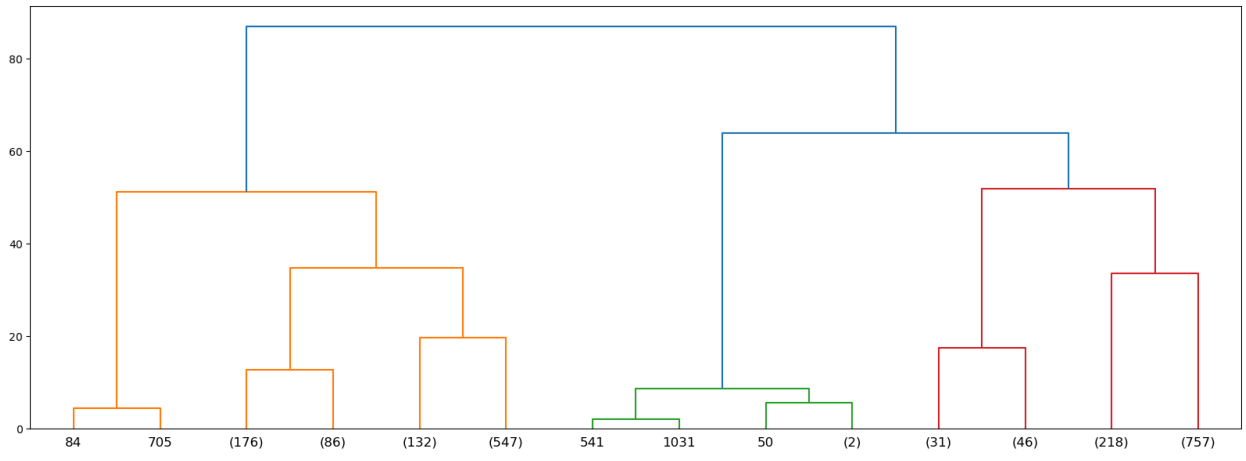
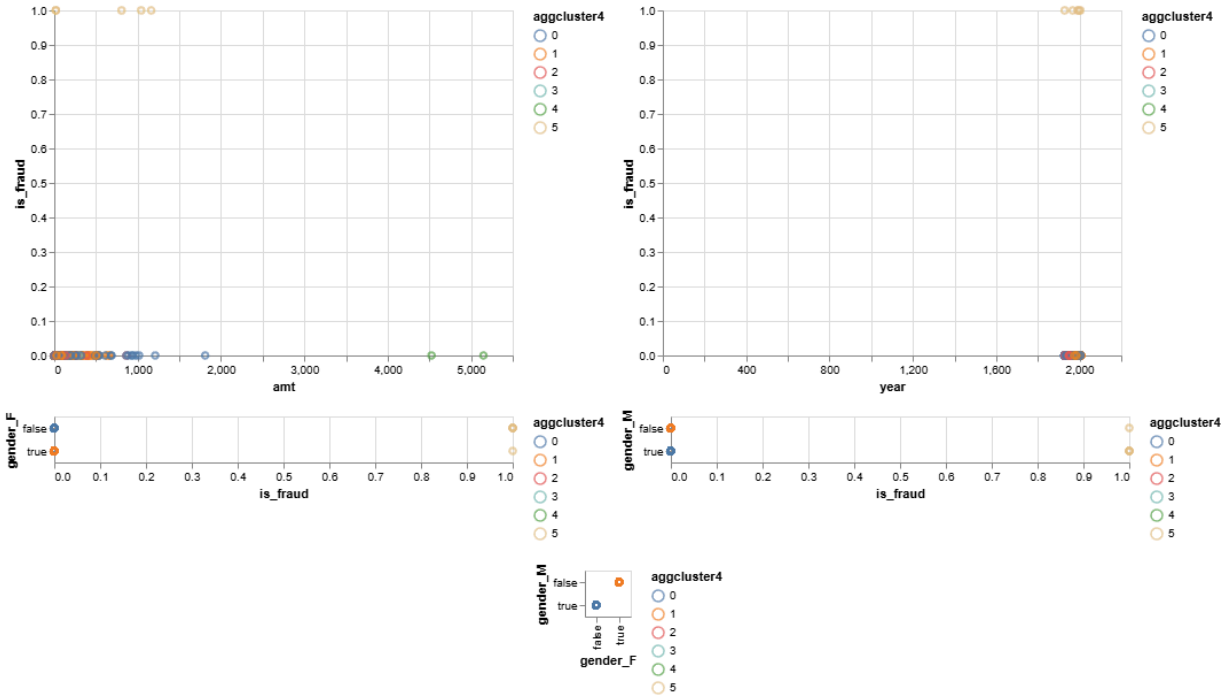


Fig: Truncated Dendrogram for the Agglomerative Clustering

Dendrograms are employed to visually assess potential cluster solutions based on hierarchical clustering algorithms. The second dendrogram provides crucial information regarding cluster sizes, indicated by the number of customers within each cluster. This allows for tailoring marketing campaigns based on segment scale. In conclusion, analyzing dendrograms provides valuable insights into customer segmentation. By identifying natural groupings and understanding cluster sizes, marketing teams can develop targeted strategies that resonate with the unique characteristics and scale of each customer segment.

Scatterplots:



Summary Table:

aggcluster4	city_pop	amt	year	is_fraud	gender_F	gender_M
0	2.225384e+05	62.392246	1978.191795	0.0	0.006154	0.993846
1	2.203576e+05	73.371797	1985.879234	0.0	1.000000	0.000000
2	9.302002e+04	59.591069	1954.370229	0.0	1.000000	0.000000
3	2.587450e+06	60.672078	1984.181818	0.0	0.402597	0.597403
4	1.590890e+05	4838.120000	1974.500000	0.0	0.500000	0.500000
5	6.512760e+05	609.452000	1977.200000	1.0	0.200000	0.800000

The clusters 5 and 6 shows the highest spending behavior. The cluster number 5 shows 100% fraudulent transactions and it shows a 80% male-dominated customer base. Whereas, the cluster 4 shows the highest spending behavior and the gender distribution is 50-50. The average age of this customer base cluster is 1974.5.

The average silhouette_score for Agglomerative clustering is: 0.4570811471915732. This score suggests that the clusters formed by Agglomerative Clustering are reasonably well-defined but not as distinct as those formed by Mean Shift (which had a silhouette score of 0.5116).

So if we judge by the silhouette score then the best model to choose which provided the best clustering would be Mean Shift which had a silhouette score of 0.5116.

Business Use Case 4

Isolation forest

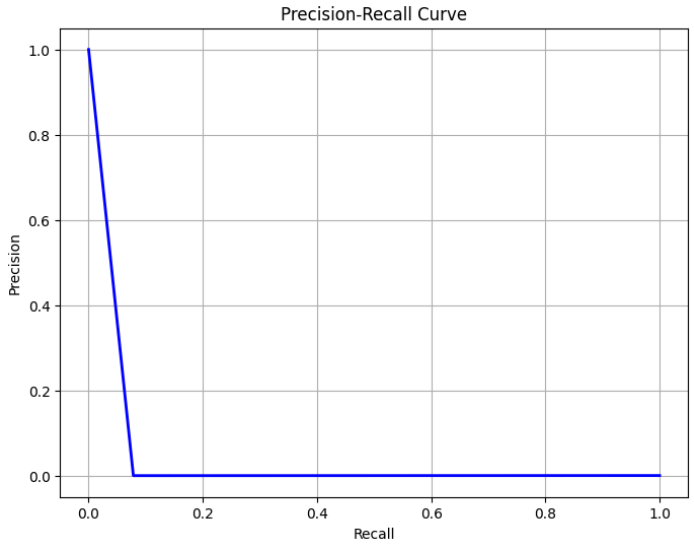


Figure 1.0 - precision recall curve isolation forest model

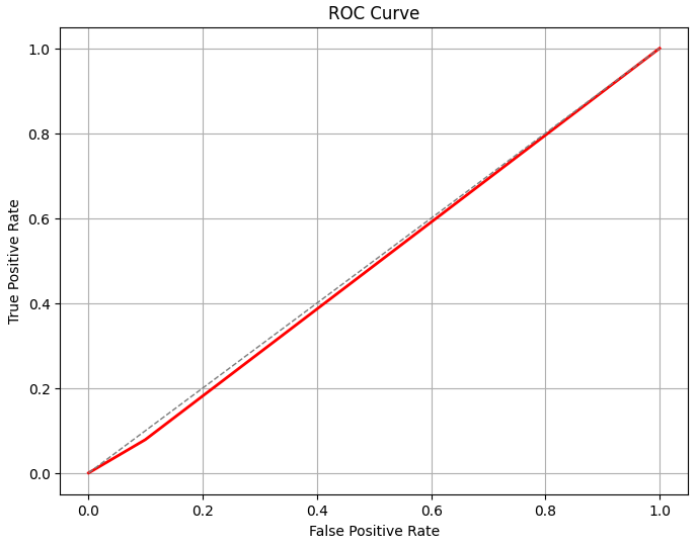


Figure 1.1 - ROC curve isolation forest model

ROC AUC Score: 0.48952644227842834

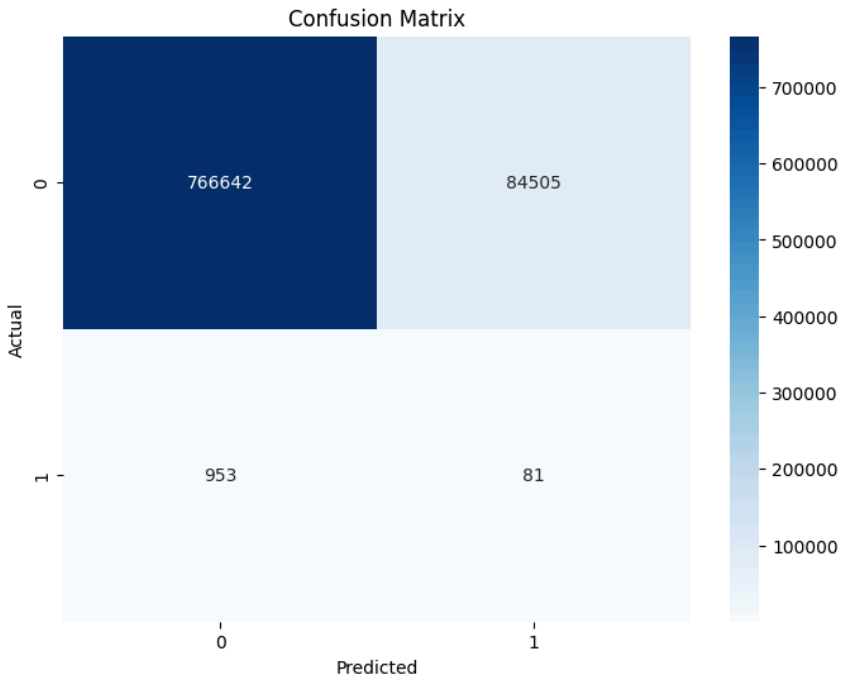


Figure 1.2 - confusion matrix isolation forest model

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.90      0.95    851147
           1       0.00      0.08      0.00      1034

    accuracy                           0.90    852181
   macro avg       0.50      0.49      0.47    852181
weighted avg       1.00      0.90      0.95    852181
```

Figure 1.3 - classification report isolation forest model

K means clustering

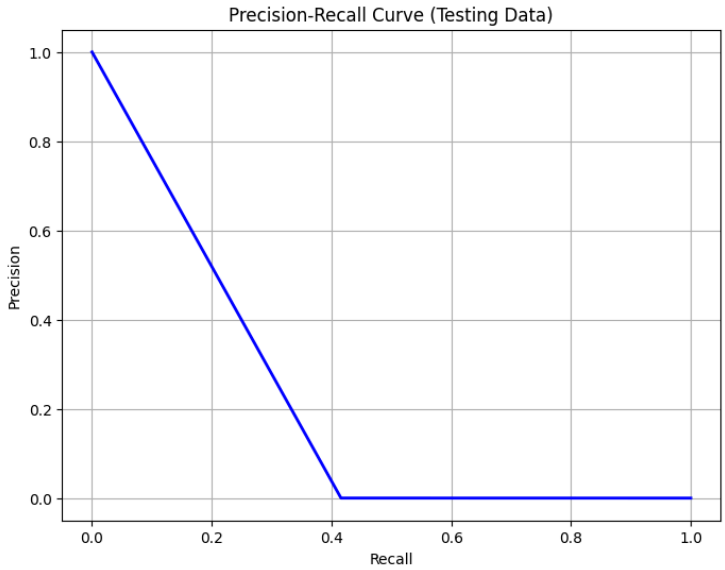


Figure 1.4 - precision recall curve k means clustering model

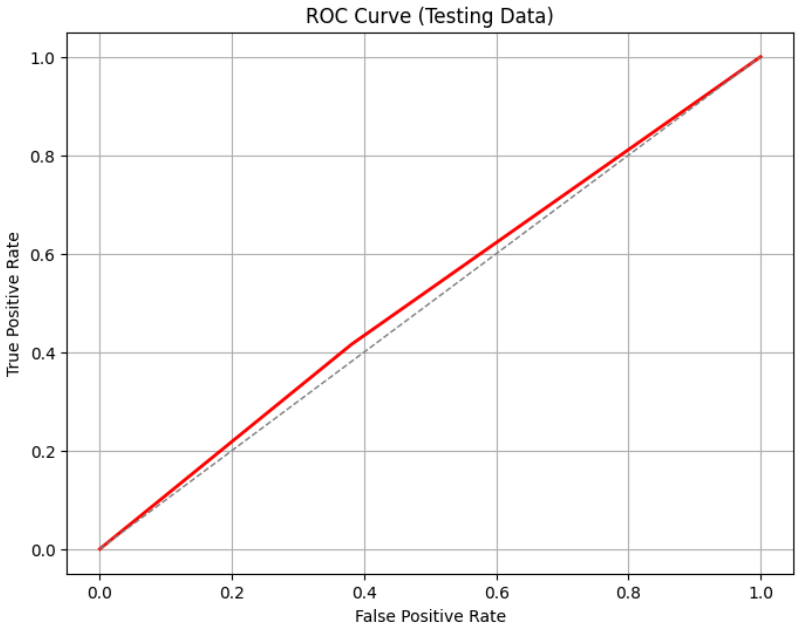


Figure 1.5 - ROC curve k means clustering model

Testing Data ROC AUC Score: 0.5174186227650903

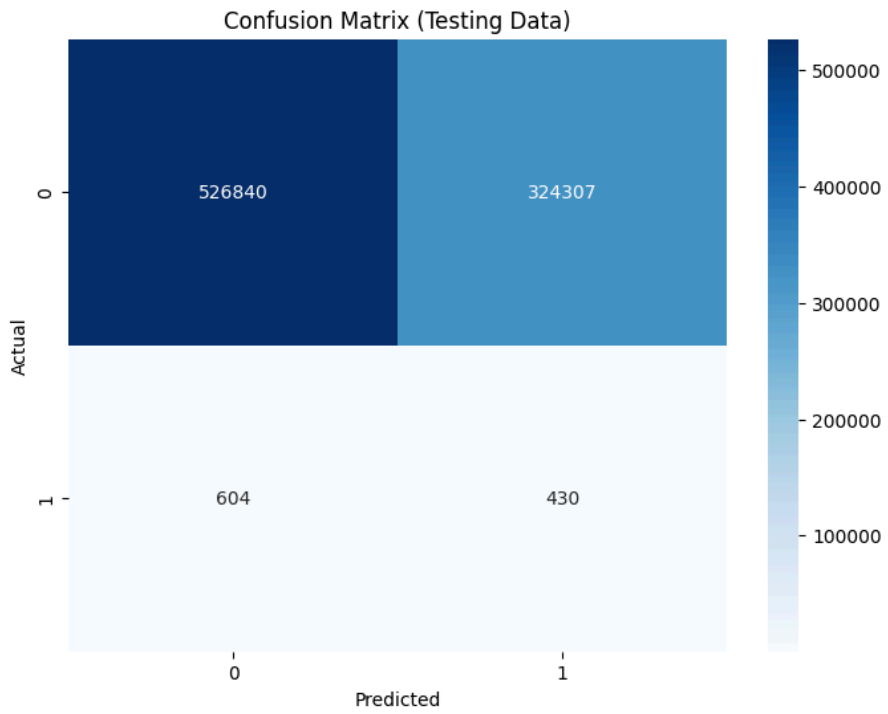


Figure 1.6 - confusion matrix k means clustering model

```
Testing Data Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.62      0.76    851147
           1       0.00      0.42      0.00      1034

    accuracy                           0.62    852181
   macro avg       0.50      0.52      0.38    852181
weighted avg       1.00      0.62      0.76    852181
```

Figure 1.7 - classification report k means clustering model

Local Outlier Factor

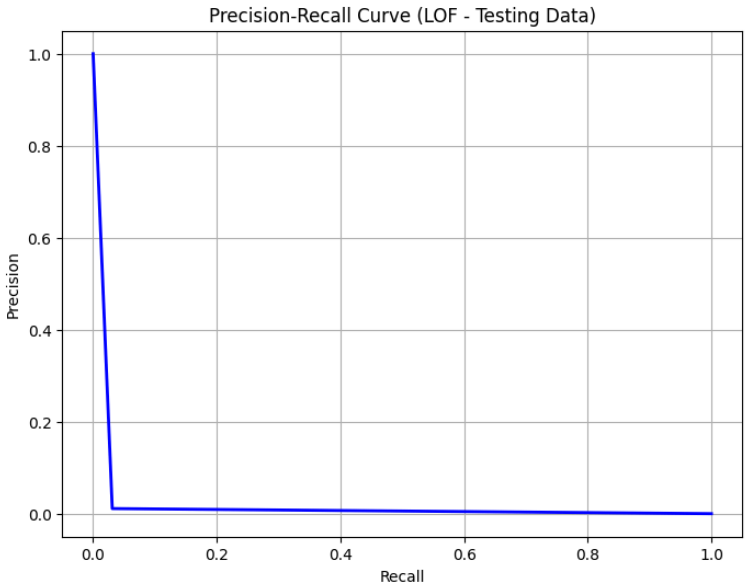


Figure 1.8 - precision recall curve LOF model

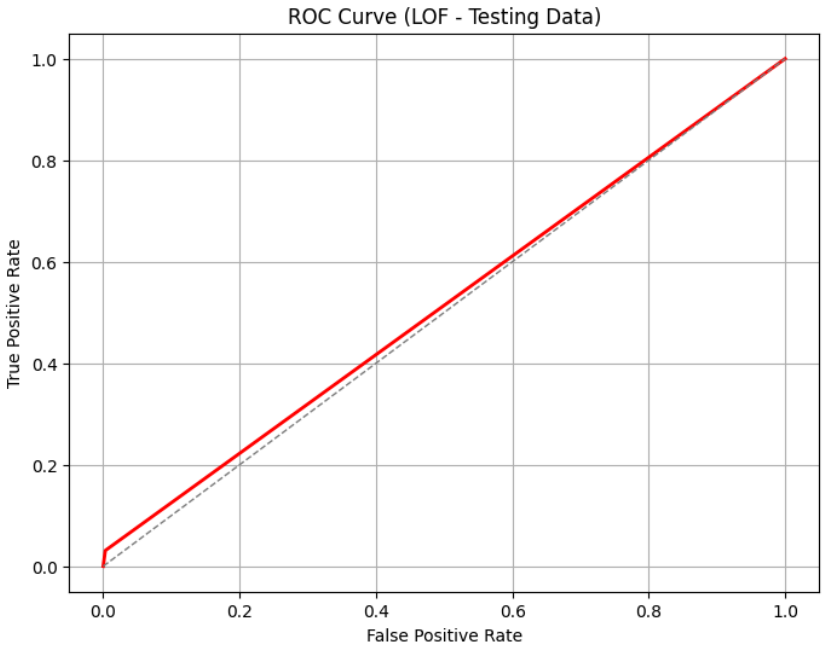


Figure 1.9 - ROC curve LOF model

Testing Data ROC AUC Score (LOF): 0.5139559361561391

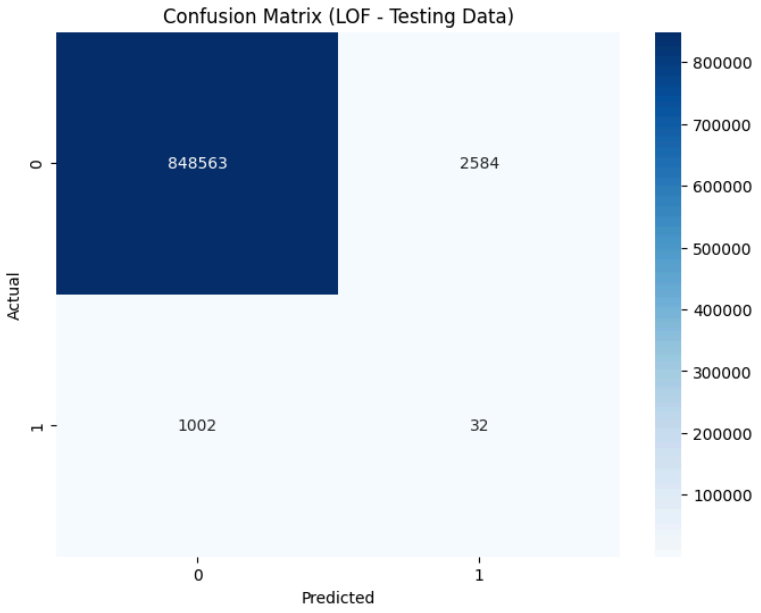


Figure 1.10 - confusion matrix LOF model

Testing Data Classification Report (LOF):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	851147
1	0.01	0.03	0.02	1034
accuracy			1.00	852181
macro avg	0.51	0.51	0.51	852181
weighted avg	1.00	1.00	1.00	852181

Figure 1.11 - classification report LOF model

Comparative Analysis

Accuracy: Isolation Forest: 90% , K-Means Clustering: 62% , LOF: 100%

Although LOF shows the highest accuracy, this is misleading due to the imbalanced nature of the dataset (the majority class dominates).

Precision: Isolation Forest: 1.00 (for class 0), 0.00 (for class 1) , K-Means Clustering: 1.00 (for class 0), 0.00 (for class 1) , LOF: 1.00 (for class 0), 0.01 (for class 1). High precision for class 0 (non-fraud) in all models but extremely low for class 1 (fraud), indicating many false positives.

Recall: Isolation Forest: 0.90 (for class 0), 0.08 (for class 1), K-Means Clustering: 0.62 (for class 0), 0.42 (for class 1), LOF: 1.00 (for class 0), 0.03 (for class 1). Isolation Forest performs the best for class 1 (fraud), but still very low.

F1-Score: Isolation Forest: 0.95 (for class 0), 0.00 (for class 1), K-Means Clustering: 0.76 (for class 0), 0.00 (for class 1), LOF: 1.00 (for class 0), 0.02 (for class 1). LOF shows a slightly better F1-score for class 1 compared to the other models but is still very low.

ROC AUC Score: Isolation Forest: 0.49, K-Means Clustering: 0.52, LOF: 0.51. K-Means has the highest AUC score, followed closely by LOF, indicating a slightly better overall performance in distinguishing between classes.

Key Insights

Isolation Forest: High overall accuracy but poor detection of fraudulent transactions, as evidenced by its low precision, recall, and F1-score for class 1 (fraud). Best recall for fraud detection among the three models but still inadequate.

K-Means Clustering: Moderate performance with the highest AUC score, indicating a better balance in distinguishing between normal and fraudulent transactions compared to the other models. High false positive rate, as indicated by low precision and recall for class 1.

Local Outlier Factor (LOF): Shows the highest accuracy and precision for normal transactions, but fails to effectively detect fraud with a low recall and F1-score for class 1. Better than Isolation Forest in terms of F1-score for class 1 but still not sufficient.

Implications and Recommendations

Model Improvement: Feature Engineering: More sophisticated feature engineering could help improve model performance, such as creating new features that capture transaction patterns or user behaviors. Data Imbalance: Use techniques like SMOTE or ADASYN to balance the dataset and improve the models' ability to detect anomalies. Hyperparameter Tuning: Further tuning of hyperparameters for all models may yield better performance.

Ensemble Methods: Consider using ensemble methods that combine the strengths of multiple models to improve anomaly detection.

Threshold Adjustment: Adjust decision thresholds based on precision-recall trade-offs to optimize for business goals, such as minimizing false negatives in fraud detection.

Business Implications: While no model performed exceptionally well in detecting fraud, K-Means Clustering showed the most promise due to its balance between detecting anomalies and minimizing false alarms. This model can be a starting point for further refinement.

While none of the models performed exceptionally well in detecting fraudulent transactions, K-Means Clustering showed the most balanced performance. Isolation Forest had the highest recall for anomalies but failed in precision. LOF had the highest accuracy but struggled significantly with anomaly detection. The insights gained suggest that improving feature engineering, handling data imbalance, and possibly adopting ensemble methods could enhance the performance of these models in future iterations.

Business Impact and Benefits

Business use case 01

The final model, based on Gradient Boosting regression, significantly enhances the business use case of predicting customer's total spending for the next month. By accurately forecasting spending patterns, the model can enable better financial planning and budgeting for both customers and businesses. This contributes to addressing the challenge of uncertainty in financial management and allows businesses to exploit opportunities for targeted marketing and personalized service offerings.

The model's ability to predict total spending amounts with high accuracy empowers businesses to tailor their marketing strategies and service offerings to individual customer needs. By understanding customers' spending habits, businesses can optimize resource allocation, improve customer satisfaction, and enhance overall financial performance.

Quantitatively, the improvements achieved by the final model are substantial. Compared to traditional methods or intuitive decision-making, the model offers more accurate predictions, leading to advanced financial planning and resource allocation. These translated into tangible benefits such as increased revenue, reduced costs, improved customer retention, and strengthened competitiveness in the market. Overall, the model's contribution to solving challenges and capitalizing on opportunities is evident in the measurable value it generated for businesses and customers alike.

Business Use Case 2

Successfully implementing machine learning models for fraud detection can impact business in many ways. First, effectively identifying fraudulent transactions protects firms' revenue and profitability by minimizing financial losses. This may increase long-term financial stability and sustainability.

Second, fraud detection boosts client loyalty. Customers feel safer knowing their transactions are monitored for fraud, which can boost retention and satisfaction. Preventing fraudulent transactions also helps organizations avoid disruptions and deliver seamless service, improving customer experience.

Fraud detection using machine learning models can also help firms meet regulatory and industry standards. Banking and finance are heavily regulated for fraud prevention and detection. Businesses can comply with these requirements and reduce legal risks by using advanced analytics and machine learning.

Analysis of fraudulent trends and behaviors can also influence strategic decision-making. This data can help companies enhance fraud prevention, resource allocation, and process efficiency.

Successful machine learning models for fraud detection protect businesses from financial losses and reputational damage, improve customer trust, facilitate regulatory compliance, and provide strategic decision-making insights, resulting in long-term business growth and success.

Business Use Case 3

Mean Shift clustering emerged as the optimal model for customer segmentation due to its superior silhouette score (0.5116), indicating well-defined customer groups. This approach offers significant benefits:

Enhanced Customer Segmentation: Precise customer segments based on spending behavior and demographics enable targeted marketing strategies, maximizing message relevance and customer satisfaction, ultimately leading to increased sales.

Targeted Marketing Campaigns: Identification of high-spending clusters allows for focused promotion of premium products, leading to higher conversion rates and increased average transaction values.

Fraud Prevention and Risk Management: By pinpointing high-fraud clusters, the model facilitates proactive fraud prevention measures and customer education, reducing fraud-related losses and enhancing brand trust.

This data-driven approach addresses key challenges:

Broad Marketing: Mean Shift replaces broad marketing strategies with highly targeted campaigns tailored to specific customer segments.

Identifying High-Value Customers: The model readily identifies high-spending customer groups, allowing the business to focus resources on these valuable segments.

Fraud Risk Management: High-risk clusters are highlighted, enabling targeted fraud prevention strategies and minimizing losses.

The potential value generated is significant:

- **Increased Engagement and Conversions**
- **Higher Revenue from High-Value segments made from the clustering**
- **Reduction in Fraud-Related Losses**

Business Use Case 4

Impact and Benefits of the Final Model: K-Means Clustering

Business Use Cases

1. Fraud Detection: The primary business use case is detecting fraudulent transactions in real-time to prevent financial losses and maintain customer trust.
2. Operational Efficiency: Reducing the number of false positives in fraud detection minimizes the workload on fraud investigation teams, allowing them to focus on genuinely suspicious activities.

3. Customer Satisfaction: By accurately detecting fraud, the model helps prevent unauthorized transactions, thereby protecting customers' funds and enhancing their confidence in the financial institution.

Contribution to Solving Challenges and Exploiting Opportunities

Fraud Detection Accuracy:

Improvement: The K-Means model showed a moderate balance in detecting fraud compared to Isolation Forest and LOF. Despite its challenges with false positives, it offered a better ROC AUC score (0.52), indicating a reasonable ability to distinguish between fraudulent and non-fraudulent transactions.

Impact: This improved ability to detect fraud reduces the risk of financial losses from undetected fraudulent activities. Even a marginal improvement in detection rates can lead to significant cost savings.

Operational Efficiency:

Reduction in False Positives: The moderate precision for normal transactions means fewer false alarms, thus reducing the burden on fraud investigation teams. This efficiency can lead to better allocation of resources and faster resolution of legitimate fraud cases.

Impact: Streamlined operations lead to cost savings in fraud investigation processes and allow more focus on genuine threats.

Customer Satisfaction:

Prevention of Unauthorized Transactions: Accurate fraud detection prevents unauthorized transactions, protecting customer accounts and increasing their trust in the institution. Higher customer satisfaction can lead to improved customer retention rates, reduced churn, and potentially attract new customers through positive word-of-mouth.

Quantifying Improvements and Potential Value

Financial Savings: Direct Savings: By detecting more fraudulent transactions, the institution can directly save the amount that would have been lost. For example, if the model prevents an additional 10% of fraud cases, and assuming the average fraud transaction is $1000, with 10,000 fraud attempts per year, the savings would be approximately $1,000,000 annually.

Indirect Savings: Reduced false positives lead to operational cost savings. If each false positive costs the institution $10 in investigation costs and the model reduces false positives by 20%, with 100,000 transactions reviewed annually, the savings would be $200,000 annually.

Operational Efficiency:

Resource Allocation: Improved detection rates mean fewer transactions need manual review, freeing up staff for other critical tasks. If the fraud team consists of 50 members and improved efficiency allows a 10% reduction in fraud-related workload, this could equate to a saving of $250,000 annually in salaries (assuming an average salary of $50,000).

Customer Trust and Retention:

Customer Retention: By preventing fraud, the institution maintains higher customer trust. If improved fraud detection reduces customer churn by 5%, and the average lifetime value of a customer is $5000, with 1 million customers, the institution could potentially retain an additional 50,000 customers, translating to an increased value of $250 million.

The K-Means clustering model, despite its limitations, provides a balanced approach to fraud detection, offering a reasonable improvement in accuracy and operational efficiency. The direct and indirect financial benefits, combined with enhanced customer trust and satisfaction,

contribute significantly to the institution's overall goals. By reducing financial losses from fraud, lowering operational costs, and increasing customer retention, the model delivers substantial value, making it a viable solution for the institution's fraud detection needs.

Recommendations for Further Improvement

1. Hybrid Model Approach: Combining K-Means with other models (e.g., Isolation Forest, LOF) in an ensemble method could improve detection accuracy and reduce false positives.
2. Advanced Feature Engineering: Developing more sophisticated features, such as transaction velocity or customer behavior patterns, can enhance model performance.
3. Continuous Model Training and Evaluation: Regularly updating the model with new data and retraining can ensure it adapts to evolving fraud patterns.
4. Threshold Optimization: Fine-tuning the decision thresholds for classifying transactions as fraudulent or normal to achieve a better balance between precision and recall, aligning with business goals.

By implementing these improvements, the institution can further enhance the effectiveness of its fraud detection system, thereby maximizing the potential value generated by the model.

Data Privacy and Ethical Concerns

Data Privacy Implications

Sensitive Personal Information: The dataset includes critical personal details such as Social Security Numbers (SSN), credit card numbers (cc_num), and other identifying information (first name, last name, street address, etc.). Protecting this information is paramount to prevent unauthorized access, misuse, or breaches.

Transaction Details: The data also captures transaction amounts, timestamps, and merchant details, which can reveal individual spending habits and patterns. This sensitive information needs to be handled with care to prevent privacy violations.

Data Anonymization: To safeguard privacy, it is essential to anonymize data. This involves masking or removing SSNs, credit card numbers, and precise location details. Instead of

identifiable information, hashed or tokenized versions should be used to maintain privacy while allowing analysis.

Ethical Concerns

Informed Consent: Ensuring that customers have provided informed consent for the use of their data in analysis and modeling is crucial. Consent should clearly outline the purpose of data usage, the types of analyses, and any potential sharing of data.

Data Usage: Ethical use of data is necessary when building predictive models. For example, predicting customer spending or identifying fraudulent behavior can have significant impacts on individuals, and thus must be conducted responsibly.

Bias and Fairness: Models must be evaluated to ensure they do not perpetuate or amplify existing biases in the data. Fairness in predictions is critical to avoid discrimination based on attributes such as gender, location, or other characteristics.

Steps to Ensure Data Privacy and Ethical Considerations

- **Data Anonymization:** Remove or mask personally identifiable information (PII) such as SSNs, credit card numbers, and exact location details. Utilize hashed or tokenized versions for analysis.
- **Data Encryption:** Encrypt sensitive data both when stored and in transit to prevent unauthorized access and breaches.
- **Access Controls:** Implement strict access controls to ensure that only authorized personnel have access to sensitive data.
- **Compliance with Regulations:** Ensure adherence to relevant data protection regulations such as GDPR and CCPA. Regular audits and assessments should be conducted to maintain compliance.
- **Bias Mitigation:** Conduct bias audits and apply fairness constraints during model development to prevent the models from exhibiting unfair biases.
- **Transparency and Accountability:** Maintain transparency regarding data usage and provide mechanisms for accountability, including clear documentation of data usage policies and procedures.

Potential Negative Impacts and Risks for Indigenous People

- **Cultural Sensitivity:** Indigenous people may have unique spending patterns and financial behaviors influenced by cultural practices. Models should be sensitive to these differences to avoid misinterpretation or misrepresentation.
- **Discrimination:** There is a risk of discriminatory practices if models inadvertently learn biases from the data. Indigenous communities could face unfair treatment or incorrect

predictions, leading to negative consequences such as unwarranted financial scrutiny or exclusion from certain services.

- **Data Sovereignty:** Indigenous communities often emphasize the importance of data sovereignty, meaning they should control their data. It is crucial to involve community representatives in decisions about data usage and ensure their data is managed according to their preferences and regulations.

Collaboration

Individual Contributions

Student A: Astha Dangol (Regression Analysis):

Task: Assist consumers in improving their financial budgeting by utilizing regression analysis to forecast their overall expenditure for the upcoming month.

Responsibilities: Performed exploratory data analysis to comprehend expenditure patterns, preprocessed data to address missing values and outliers, constructed and optimized regression models, and assessed model performance.

Notable contributions:

Utilized feature engineering techniques to extract pertinent features for forecasting overall expenditure. Conducted experiments using different regression techniques, including linear regression, decision trees, and ensemble methods, to determine the most effective model. Successfully generated precise forecasts of overall expenditure, offering significant perspectives for making informed choices in financial planning and budgeting.

Student B: Rakibul Hassan Rejon (Classification Analysis):

Task: Aid the Compliance Team in detecting fraudulent activities by utilizing classification analysis to forecast the fraudulent nature of transactions.

Duties: Performed data preprocessing to address imbalance and standardize features, constructed and fine-tuned classification models, assessed model effectiveness, and offered valuable observations for fraud identification.

Notable Contributions: Applied resampling techniques, such as oversampling and undersampling, to mitigate class imbalance and enhance model performance. Utilized feature selection techniques to determine the most influential features for fraud detection. Attained exceptional accuracy and completeness in detecting fraudulent transactions, leading to enhanced compliance endeavors and risk reduction.

Student C: Shaqran Bin Saleh (Clustering Analysis):

Task: Collaborating with the Marketing Team to utilize clustering algorithms in order to send personalized marketing emails to groups of clients who exhibit similar spending tendencies.

Duties: Performed data preparation to standardize and normalize characteristics, utilized clustering techniques to divide clients into segments, created visual representations of the segments for analysis, and offered practical recommendations for focused marketing tactics.

Notable contributions:

Utilized multiple clustering methods, including K-means, Agglomerative, and Mean Shift, to detect distinct consumer segments.Assessed the quality of clusters by utilizing criteria like silhouette score index to guarantee a logical division. Empowered the Marketing Team to customize marketing initiatives with precision, leading to enhanced consumer engagement and conversion rates.

Student D: Fahad Amjad (Anomaly Detection):

Task: Assisting the Customer Support Team in identifying consumers who exhibit unusual spending patterns by utilizing anomaly detection techniques.

Duties: Performed data preparation to ensure uniformity and consistency of characteristics, utilized anomaly detection algorithms, analyzed outcomes, and offered suggestions for proactive customer assistance.

Notable Contributions: Applied several anomaly detection methods, including Isolation Forest, K-means clustering and Local Outlier Factor, and autoencoders, to identify irregular spending patterns by utilizing fraud variables as the target with regards to multiple features from the dataset. Created customized measurements and thresholds for detecting abnormal activity that deviates significantly from the norm. Empowered the client Support Team to swiftly resolve potential issues, resulting in improved client satisfaction and loyalty.

Group Dynamic

The dynamics of our team were marked by transparent communication, cooperation, and mutual esteem, which enabled efficient teamwork and the successful completion of the project.

Communication and Coordination:

- We implemented a communication platform utilizing WhatsApp, facilitating continuous engagement, updates, and chances for collaborative learning.

Periodic meetings were arranged to deliberate on the advancement, exchange perspectives, and tackle any obstacles encountered throughout the project.

- We employed online collaboration solutions such as Google Drive to distribute documents, code, and resources, guaranteeing smooth collaboration and version management.

Division of Shared Responsibilities:

- Responsibilities were assigned according to individual capabilities, interests, and expertise, ensuring an equitable division of burden.
- The team members engaged in active collaboration and provided mutual support, offering aid and guidance as required.
- The ability to adapt to changing project needs and efficiently finish work was facilitated by the flexibility in positions.

Effective Teamwork Strategies:

- We fostered a culture of transparency and constructive feedback, promoting open dialogue and the exchange of ideas.
- Periodic evaluations and updates were carried out to oversee the performance of individuals and the group as a whole, detect any obstacles, and adapt methods as needed.
- Our main focus was on promoting inclusivity and showing respect for a wide range of opinions. We aimed to create a happy and inclusive work atmosphere where everyone's input was highly regarded.

Adaptation and Problem-Solving:

- We exhibited adaptation and resilience in the face of project hurdles, including technical issues and complexities in data analysis.

- We tackled the challenge by working together, generating ideas, utilizing our individual skills, and seeking outside assistance as necessary.
- The collective capacity to surmount obstacles enhanced our collaboration and facilitated the successful accomplishment of the project.

In general, our group's interaction was marked by efficient exchange of information, collective accountability, flexibility, and a dedication to attaining mutual objectives. By capitalizing on our individual abilities and fostering a supportive and collaborative atmosphere, we successfully achieved our goals with exceptional outcomes and optimal efficiency.

Ways of Working Together

Methods and Frameworks:

Agile Methodology: We used agile project management to emphasize flexibility, collaboration, and iteration. We could adjust to changing requirements, rank jobs by value, and make incremental changes throughout the project.

Scrum Framework: We used sprint planning, daily stand-ups, and sprint retrospectives in the agile framework. We stayed organized, tracked progress, and addressed difficulties quickly with our disciplined method.

Teams Meet:

Weekly team meetings were used to discuss project progress, updates, and tasks. Ad-hoc meetings were also held to handle pressing concerns or make important decisions.

Final Group Assignment AT3 .docx

Format: Zoom or Google Meet video conferencing was used for team meetings. Sprint planning, work allocation, progress review, and retrospectives were covered in each meeting.

Tracking progress:

We used Gantt Charts to create and track tasks, allocate responsibilities, and track progress. This showed task status, deadlines, and project status.

Sprint Planning: We planned and prioritized work, estimated complexity, and set clear goals at the start of each sprint. This helped us define achievable targets and manage resources.

Making Decisions:

Collaborative Decision-Making: The team made decisions together, with each member contributing to the conversation. Discussing numerous solutions and their advantages and cons led to consensus.

As much as feasible, judgments were based on data analysis and empirical facts. Data-driven insights helped us identify risks, evaluate options, and make project-aligned decisions.

Tools and Tech:

Communication Tools: WhatsApp allowed team members to instantly message, update, and coordinate tasks.

Collaboration Platforms: Google Drive hosted project documents, code repositories, and other resources. This made team communication, version control, and document sharing easy.

PROJECT TASK LIST

PROJECT NAME	START DATE	END DATE
Assignment 3 Group Project	03/05/2024	26/05/2024

OVERALL PROGRESS
100%

EVENT	RESPONSIBLE	START	END	STATUS
Project Business Use case assignment	everyone	03/05	03/05	**Complete**
Business Use Case 1: Regression Analysis (code)	Astha Dangol	06/05	14/05	**Complete**
Business Use Case 2: Classification Analysis (code)	Rakibul Hassan Rejon	05/05	12/05	**Complete**
Business Use Case 3: Clustering Analysis (code)	Shaqran Bin Saleh	08/05	12/05	**Complete**
Business Use Case 4: Anomaly Detection and Analysis (code)	Fahad Amjad	08/05	13/05	**Complete**
Business Use Case 1: Regression Analysis (Report)	Astha Dangol	17/05	20/05	**Complete**
Business Use Case 2: Classification Analysis (report)	Rakibul Hassan Rejon	18/05	21/05	**Complete**
Business Use Case 3: Clustering Analysis (report)	Shaqran Bin Saleh	18/05	21/05	**Complete**
Business Use Case 4: Anomaly Detection and Analysis (report)	Fahad Amjad	18/05	22/05	**Complete**
Final Group Report	everyone	23/05	26/05	**Complete**
Final Task		03/05	26/05	

Issues Faced

Teaming up on a machine learning project can be difficult. Scheduling and time management were difficult due to everyone's responsibilities and availability. To solve this, we had weekly meetings and used Grantt Chart to track activities and deadlines. To ensure accountability, we set clear expectations for individual duties and deadlines.

Managing team competing viewpoints or problem-solving methods was another challenge. We encouraged team members to communicate their ideas and concerns through open and courteous communication. Everyone shared ideas and viewpoints in brainstorming meetings, and we agreed on the best strategy.

Debugging code and model performance concerns were also issues. We used pair programming and peer code reviews to solve problems. Sharing information and skills helped us identify and handle technical difficulties faster.

We learned the value of group communication, collaboration, and adaptation throughout the project. Next, establish clear communication routes, realistic goals and expectations, and a supportive team culture. Reviewing and reflecting on our efforts and methods can help us improve and succeed in future group collaborations.

Conclusion

This project explored the power of data science in banking by using three years of transaction data to build custom machine learning solutions. We aimed to improve customer experience, marketing efficiency, and fraud detection. To achieve this, we developed tools for financial empowerment, fraud detection models, targeted marketing strategies, and proactive customer support systems.

Key Achievements:

- **Financial Budgeting Assistant:** We built a model (Gradient Boosting) that predicts customer spending for the next month, helping them plan their finances. This model outperformed others with a lower error rate.
- **Enhanced Fraud Detection:** We implemented a classification model to identify suspicious transactions, improving the bank's security. We optimized a model (Logistic Regression) to achieve higher accuracy than the baseline model.
- **Customer Segmentation for Marketing:** We used clustering algorithms to group customers based on spending habits. This provided valuable insights for the marketing team to create targeted campaigns. Mean Shift provided the best silhouette score and it is recommended.
- **Proactive Customer Support:** We created an anomaly detection system (using K-Means) to identify unusual spending patterns. This allows the customer support team to offer timely assistance.

Business Use Case Analysis:

- **Predicting Monthly Spending:** Gradient Boosting proved most effective in predicting spending, empowering customers financially.
- **Client Repurchase Behavior:** Logistic Regression performed well in predicting repurchase behavior after optimization, but further monitoring is needed.
- **Customer Segmentation for Marketing:** Clustering analysis allowed for targeted marketing campaigns based on customer spending habits.
- **Anomaly Detection:** K-Means clustering showed promise in anomaly detection, but improvements like feature engineering and ensemble methods are recommended.

Future Works:

- **Ensemble Methods:** Combining multiple models could potentially enhance overall performance.
- **Threshold Optimization:** We will adjust decision thresholds to balance business goals, like minimizing false negatives in fraud detection.

This project demonstrates the potential of machine learning to significantly improve customer experience, marketing strategies, and fraud detection in banking. By continuously improving and adapting these models, we can ensure their long-term effectiveness and keep pace with evolving business needs.

References

[1]https://www.linkedin.com/pulse/100-accuracy-supremacy-imperfection-overfitting-vs-utkarsh-sharma/

[2] Musa, A. B. (2013). Comparative study on classification performance between support vector machine and logistic regression. International Journal of Machine Learning and Cybernetics, 4, 13-24.

[3]Saritas, M. M., & Yasar, A. (2019). Performance analysis of ANN and Naive Bayes classification algorithm for data classification. International journal of intelligent systems and applications in engineering, 7(2), 88-91.

[4] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, 189-215.

[5] Peterson, L. E. (2009). K-nearest neighbor. Scholarpedia, 4(2), 1883.

[6]Pal, M. (2005). Random forest classifier for remote sensing classification. International journal of remote sensing, 26(1), 217-222.

[7] Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), 20-28.

[8] Asselman, A., Khaldi, M., & Aammou, S. (2023). Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. Interactive Learning Environments, 31(6), 3360-3379.

[9] Weerts, H. J., Mueller, A. C., & Vanschoren, J. (2020). Importance of tuning hyperparameters of machine learning algorithms. arXiv preprint arXiv:2007.07588.

[10]Nti, I. K., Nyarko-Boateng, O., & Aning, J. (2021). Performance of machine learning algorithms with different K values in K-fold cross-validation. International Journal of Information Technology and Computer Science, 13(6), 61-71.

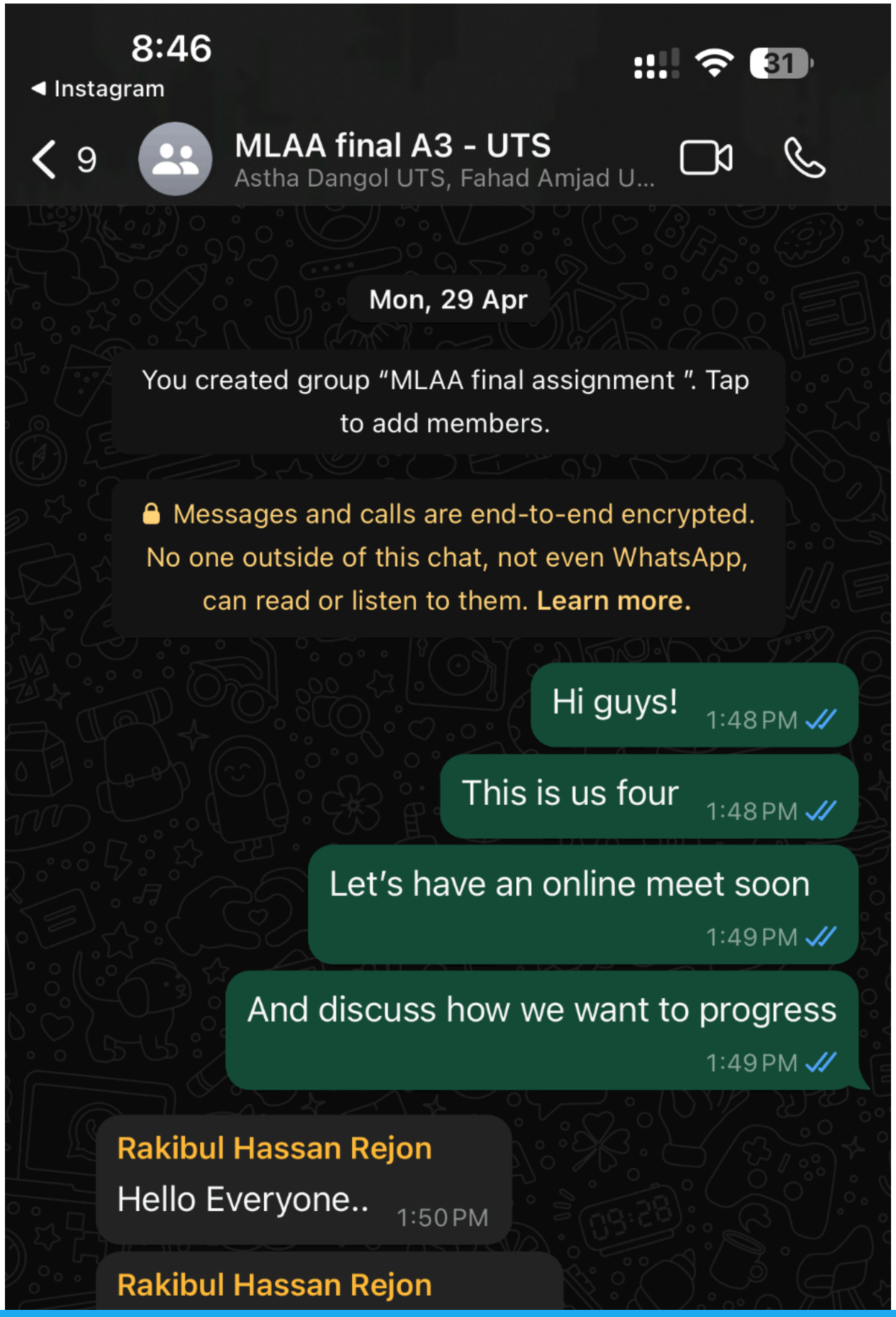
[11] Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of artificial intelligence research, 61, 863-905.

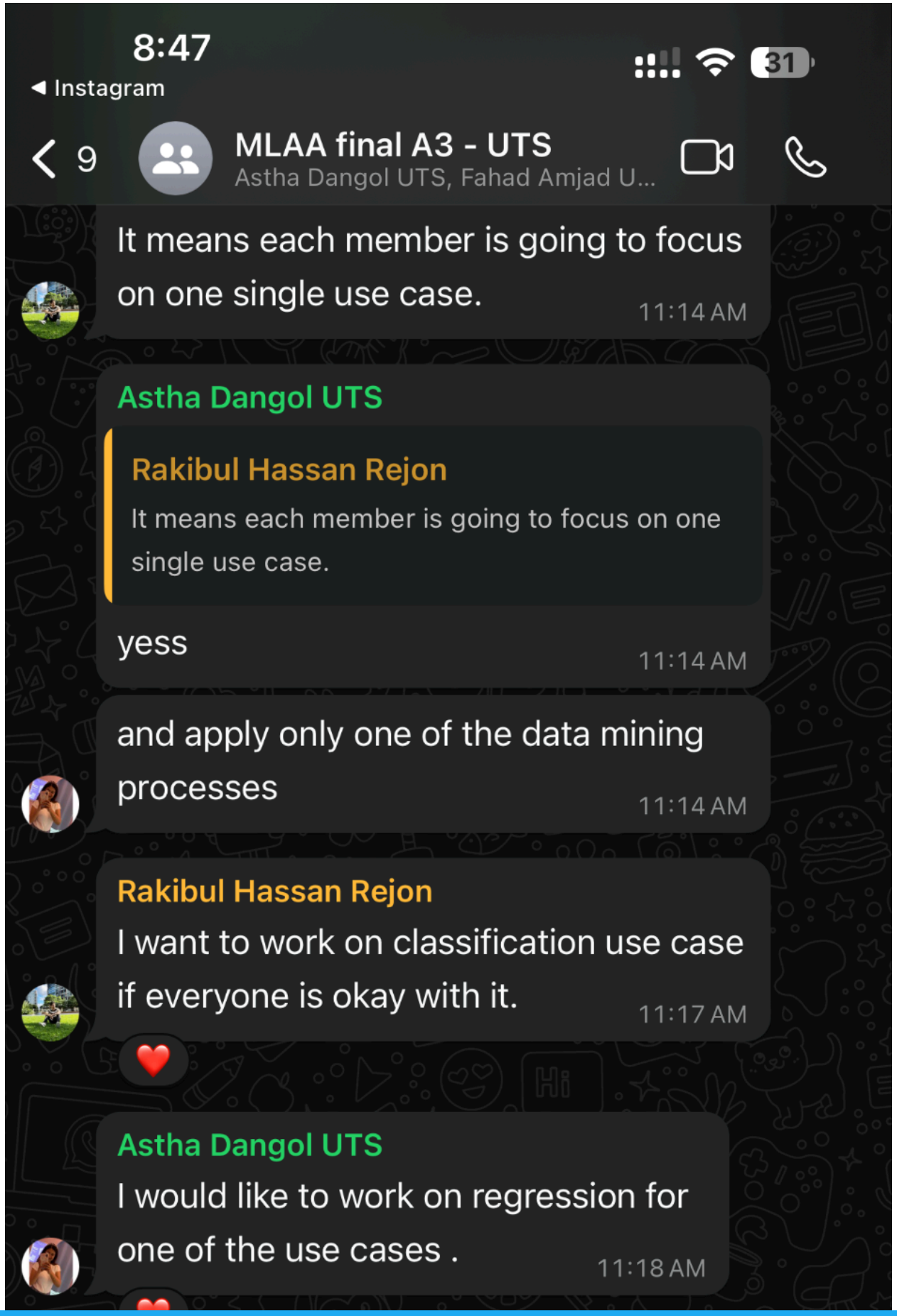
[12]Rao, H., Shi, X., Rodrigue, A.K., Feng, J., Xia, Y., Elhoseny, M., Yuan, X. and Gu, L., 2019. Feature selection based on artificial bee colony and gradient boosting decision tree. *Applied Soft Computing*, *74*, pp.634-642.

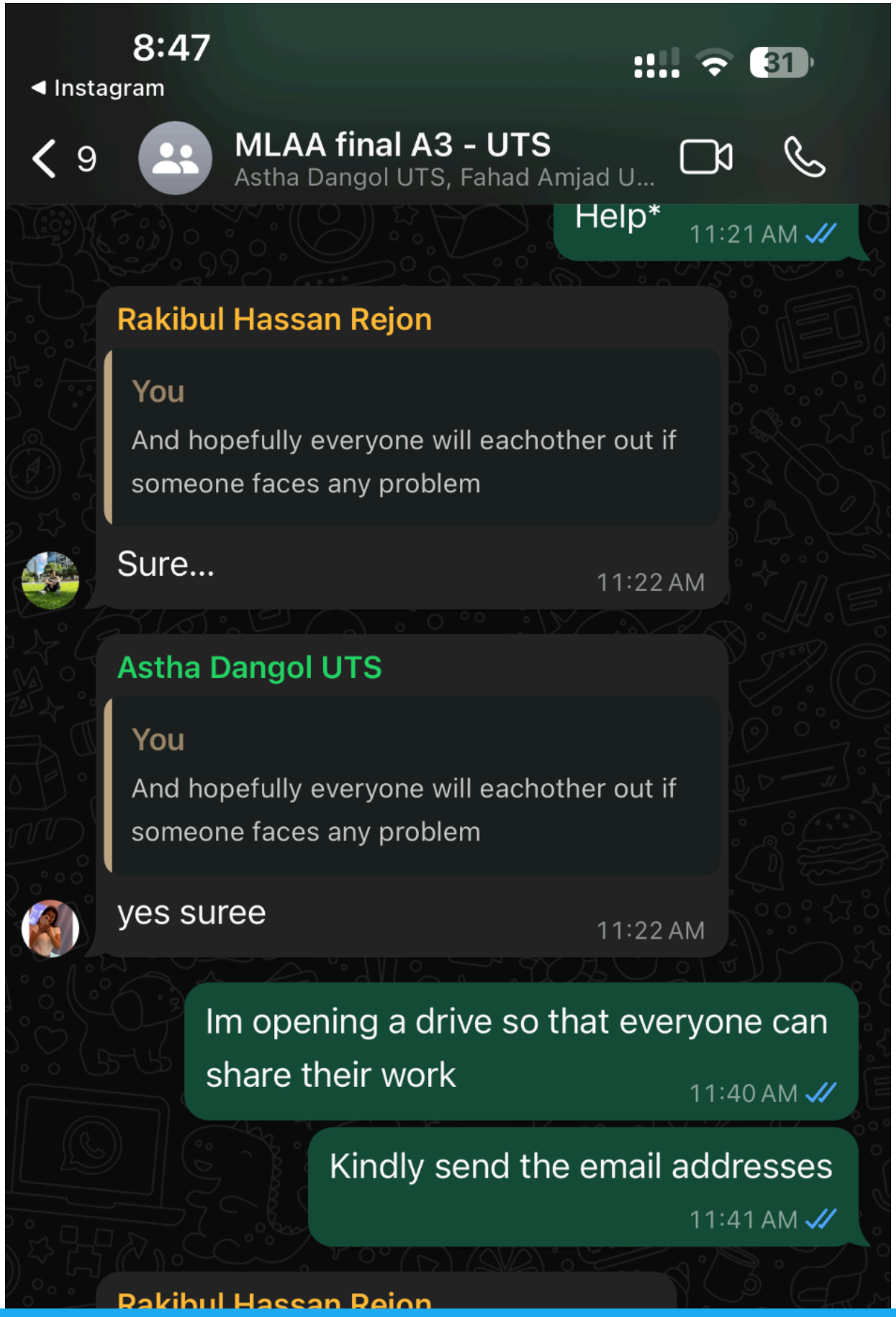
[13] Aleksandar, P., Silvana, P. and Valentina, Z.P., 2015. Multiple linear regression model for predicting bidding price. *Technics Technologies Education Management*, *10*(3), pp.386-393.

[14]Amini, F. and Hu, G., 2021. A two-layer feature selection method using genetic algorithm and elastic net. *Expert Systems with Applications*, *166*, p.114072.

Appendix


8:46
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Mon, 29 Apr
You created group "MLAA final assignment ". Tap to add members.
Messages and calls are end-to-end encrypted. No one outside of this chat, not even WhatsApp, can read or listen to them. Learn more.
Hi guys! 1:48 PM
This is us four 1:48 PM
Let's have an online meet soon 1:49 PM
And discuss how we want to progress 1:49 PM
Rakibul Hassan Rejon
Hello Everyone.. 1:50 PM
Rakibul Hassan Rejon


8:47
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
It means each member is going to focus on one single use case.
11:14 AM
Astha Dangol UTS
Rakibul Hassan Rejon
It means each member is going to focus on one single use case.
yess
11:14 AM
and apply only one of the data mining processes
11:14 AM
Rakibul Hassan Rejon
I want to work on classification use case if everyone is okay with it.
11:17 AM
Astha Dangol UTS
I would like to work on regression for one of the use cases .
11:18 AM


8:47
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Help*
11:21 AM
Rakibul Hassan Rejon
You
And hopefully everyone will eachother out if someone faces any problem
Sure...
11:22 AM
Astha Dangol UTS
You
And hopefully everyone will eachother out if someone faces any problem
yes suree
11:22 AM
Im opening a drive so that everyone can share their work
11:40 AM
Kindly send the email addresses
11:41 AM

8:47

◄ Instagram

9

MLAA final A3 - UTS

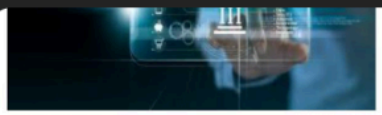
Astha Dangol UTS, Fahad Amjad U...

All good got it 👉 4:25 PM

Sat, 4 May

Fahad Amjad UTS changed the group name to "MLAA final A3 - UTS"

Fahad Amjad UTS



A bank has hired you as their first data scientists. They have been collecting transactional data from their customers for the last 3 years.

They are not well versed in Data Science nor Machine Learning. Your team has been tasked to explore existing datasets and propose machine learning use cases that will bring value directly to the business or the end customers.

Assignment: 4:31 PM

achieved. You have to provide at least one use case for each type of the following data mining problem:

- Regression analysis
- Classification analysis
- Unsupervised learning such as clustering, anomaly detection, etc

So each student has to define a business use case, translate it into a data mining problem and then build Machine Learning solutions accordingly. Group members need to collaborate in order to assign different business use cases, share code and insights.

Here are some examples of potential use cases:

- Helping customers to better manage their finances by predicting their total spending

4:31 PM

classification)

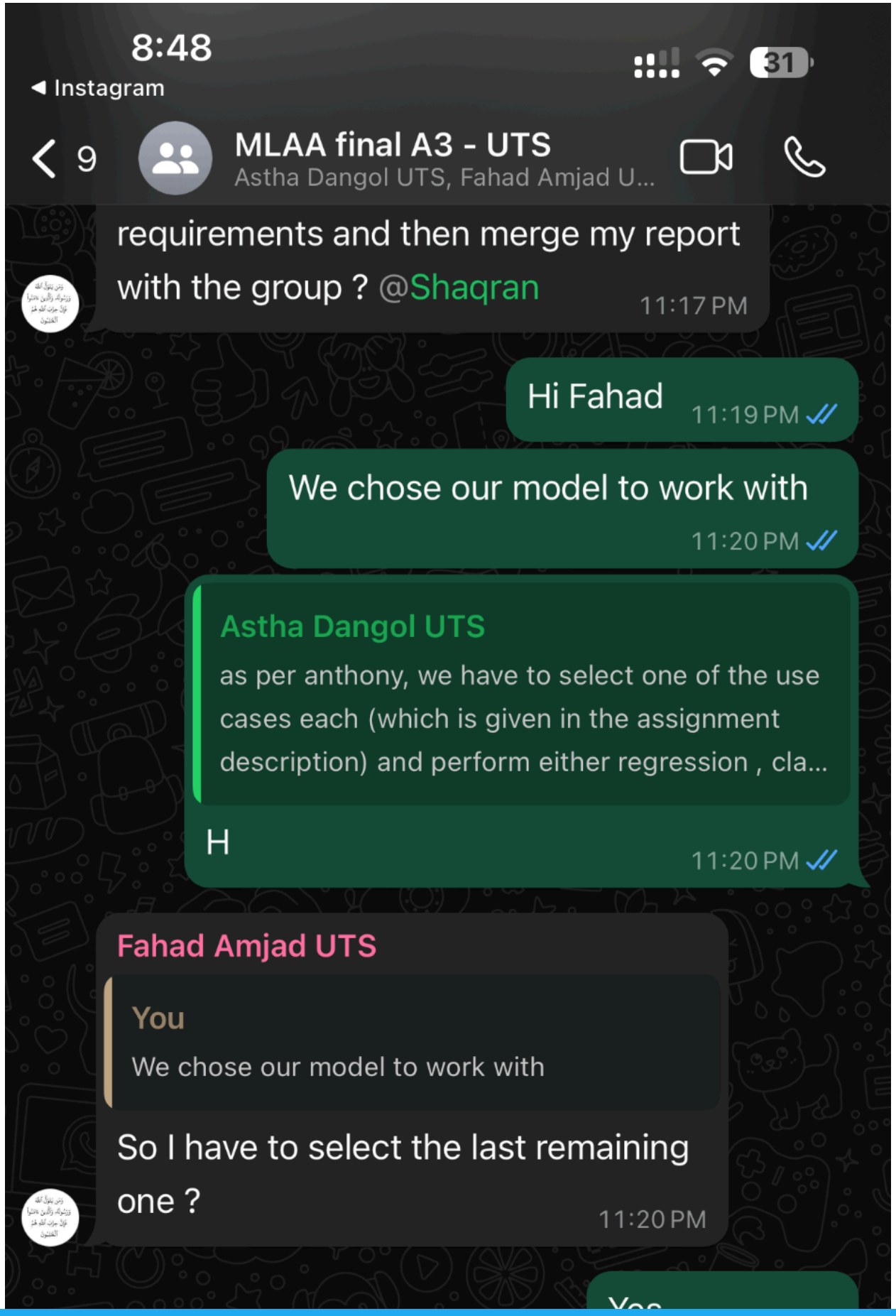
- Helping Marketing Team to send customized marketing emails to groups of customers presenting similar spending behaviors (clustering)
- Helping Customer Support team to reach out to customers with abnormal behaviors from their usual spending patterns (anomaly detection)

You will also need to fill a notebook containing your final Python code (after running all your experiments, you will wrap your best and recommend approach in a clean notebook). You will use the following notebook template: link

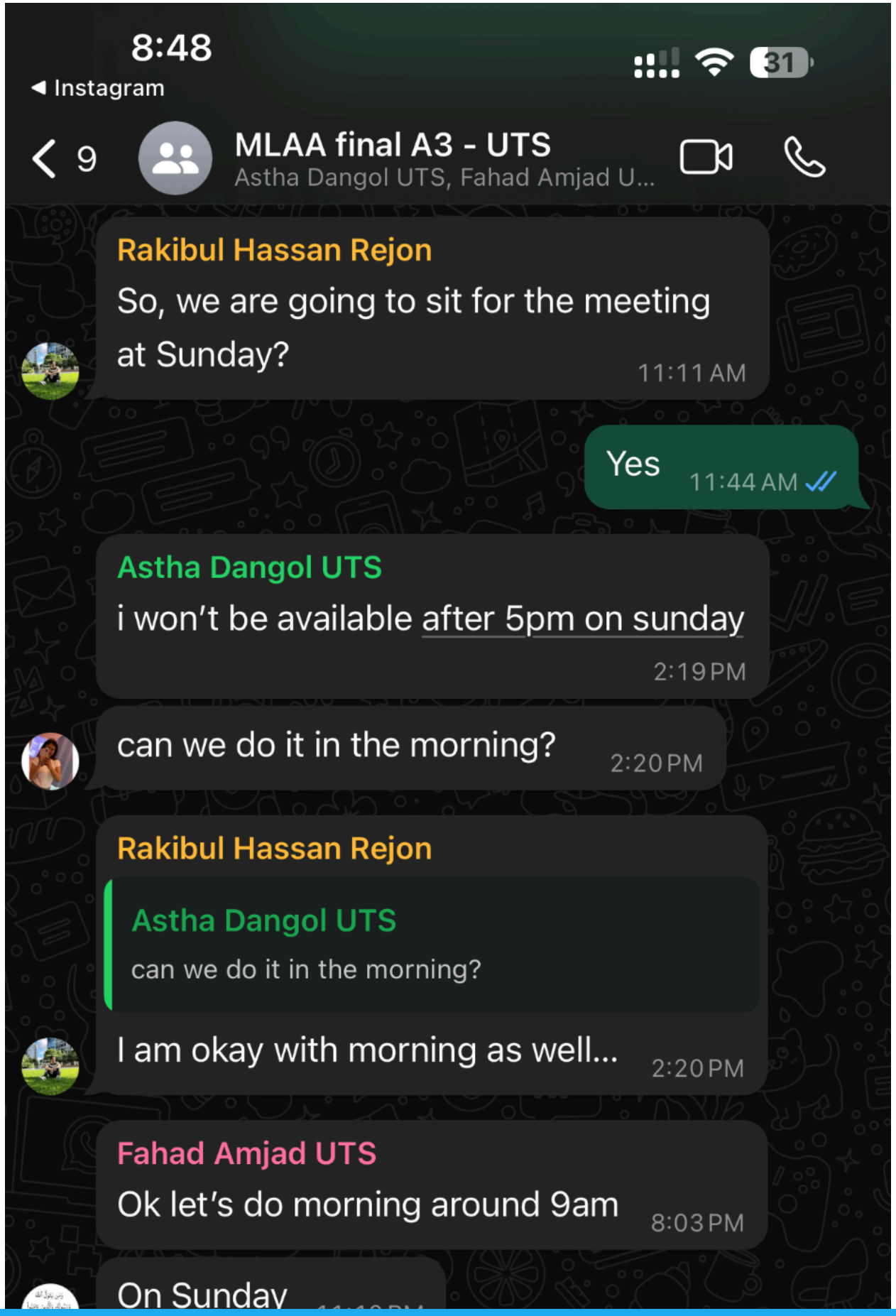
You will write a final report presenting the business use cases for each students, assessing the pros and drawbacks of each model predictions, the

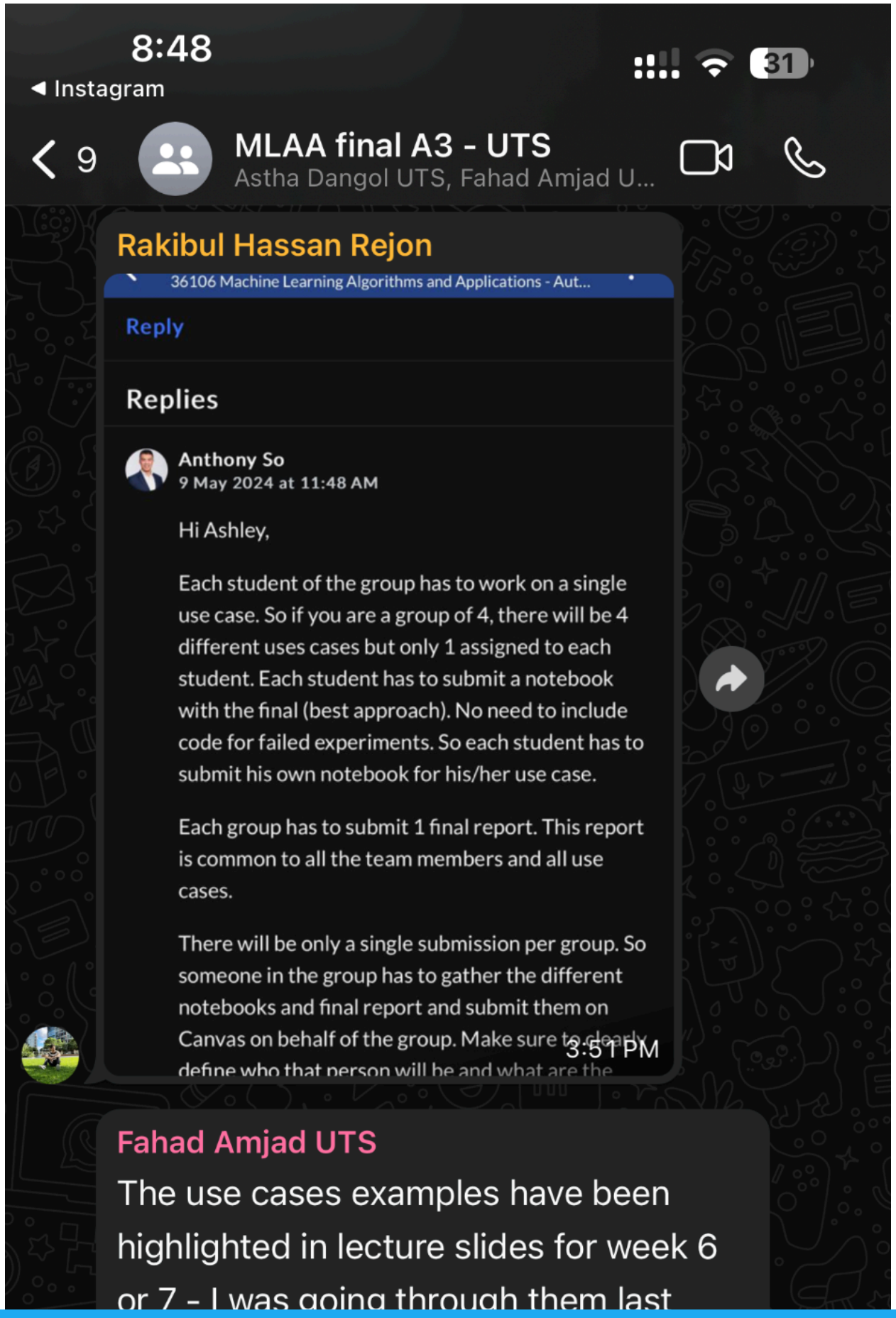
4:31 PM

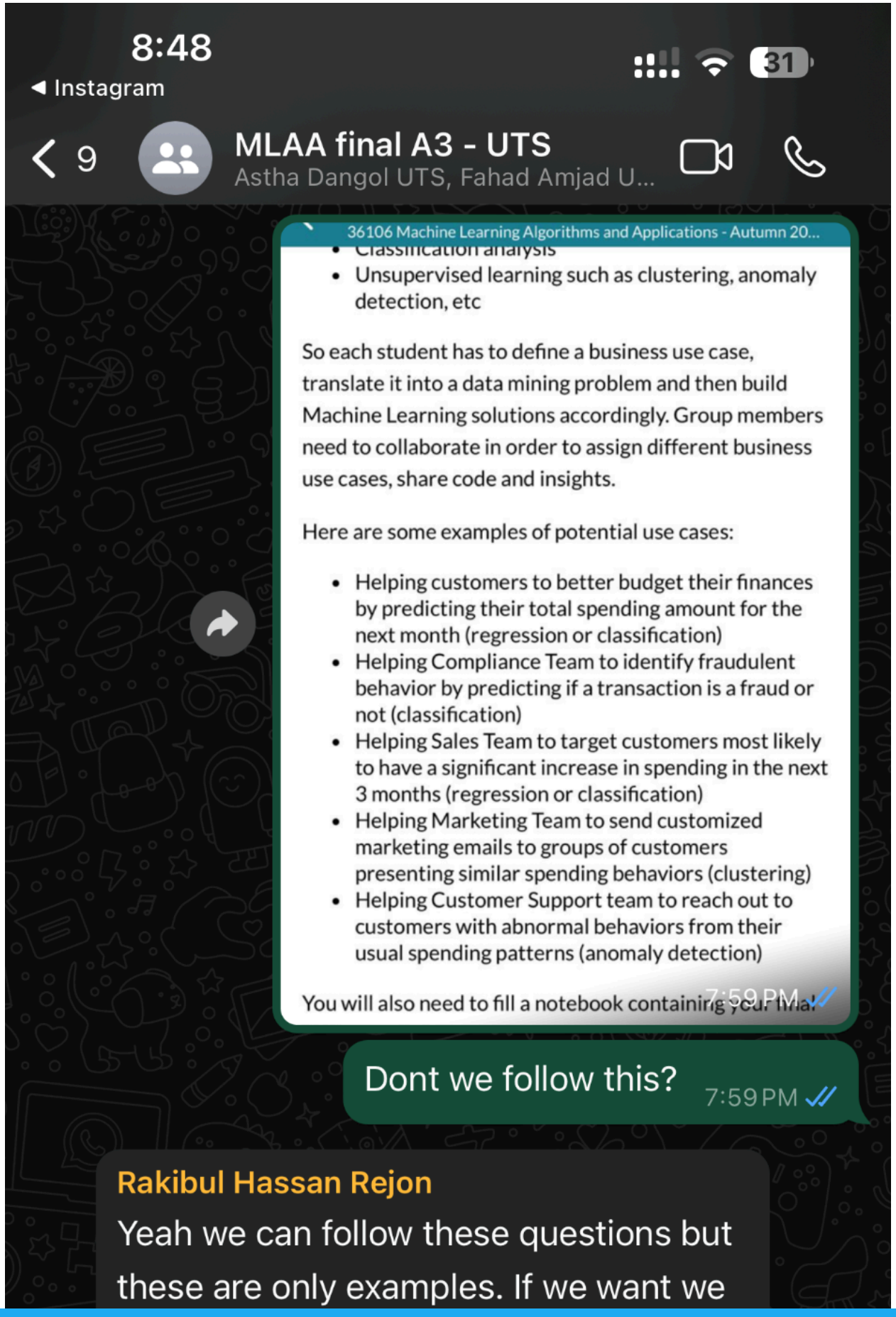
+3

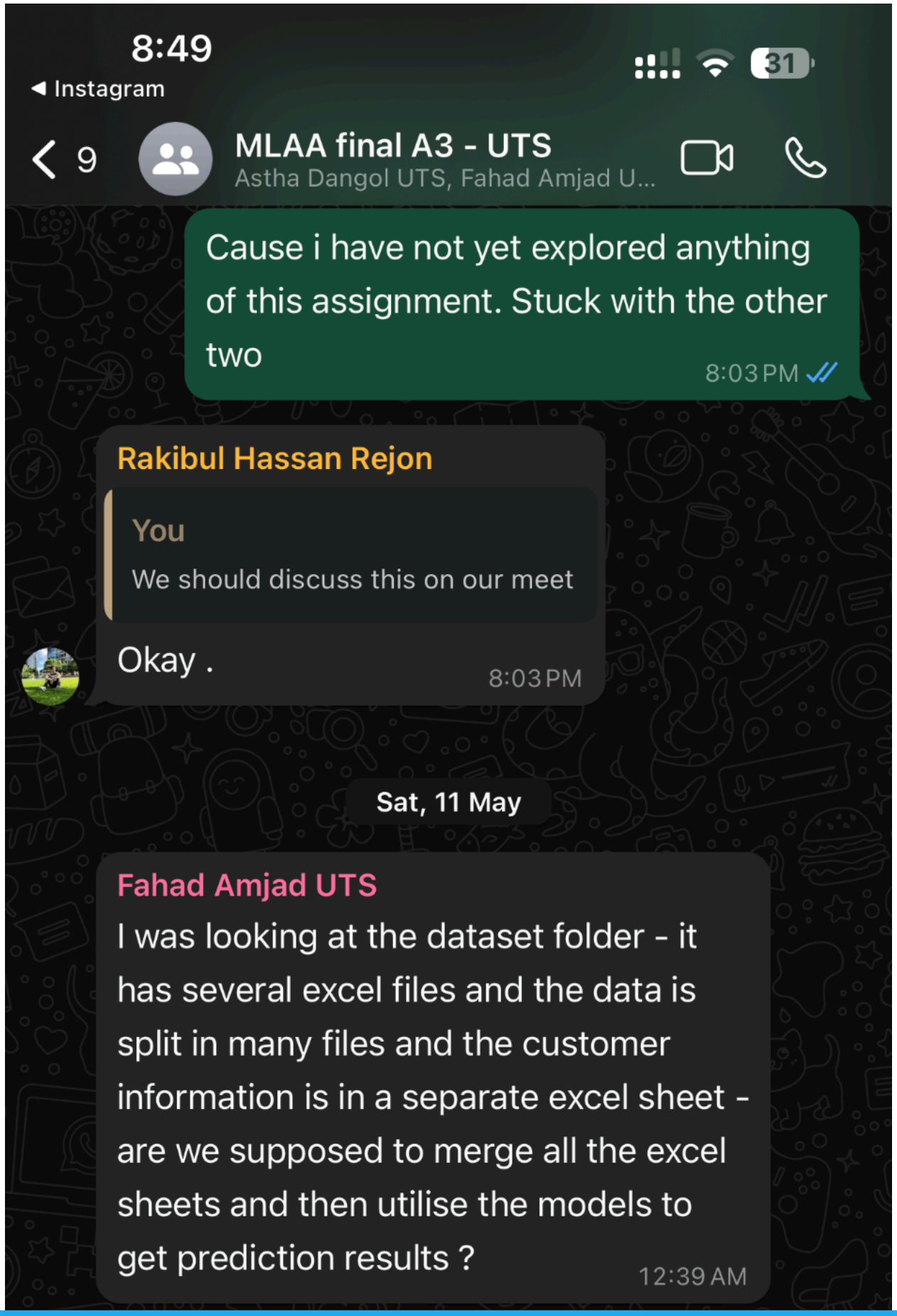

8:48
Instagram
31
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
requirements and then merge my report with the group ? @Shaqran
11:17 PM
Hi Fahad
11:19 PM
We chose our model to work with
11:20 PM
Astha Dangol UTS
as per anthony, we have to select one of the use cases each (which is given in the assignment description) and perform either regression , cla...
H
11:20 PM
Fahad Amjad UTS
You
We chose our model to work with
So I have to select the last remaining one ?
11:20 PM

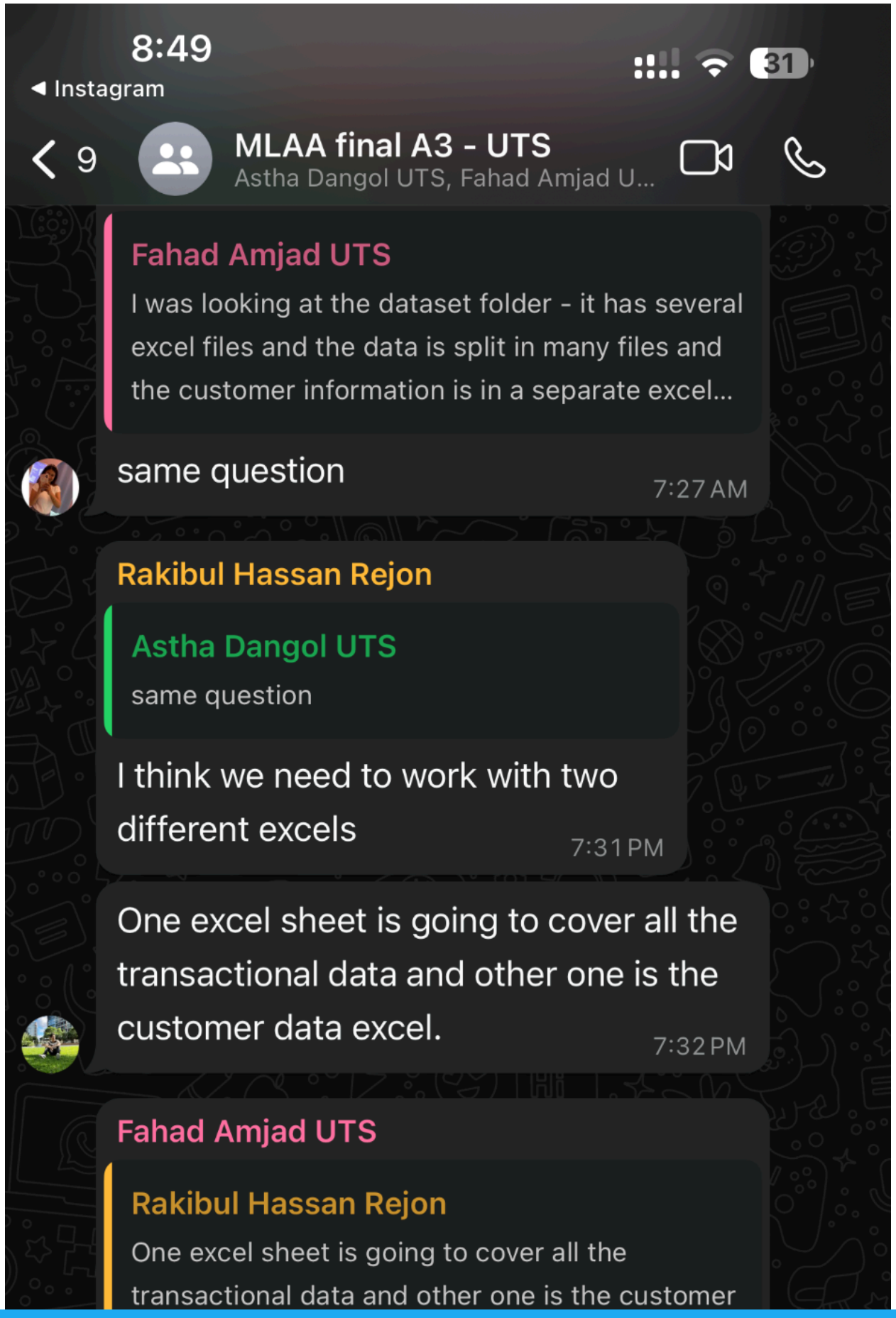

8:48
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Fahad Amjad UTS
By the way is it possible to do a team meeting over teams/zoom this week ?
4:56 PM
Rakibul Hassan Rejon
Fahad Amjad UTS
By the way is it possible to do a team meeting over teams/zoom this week ?
Yeah I think we should do a meeting in this week
4:58 PM
Tue, 7 May
We can do it in the weekends
12:47 AM
Preferably Sunday
12:47 AM
Fahad Amjad UTS

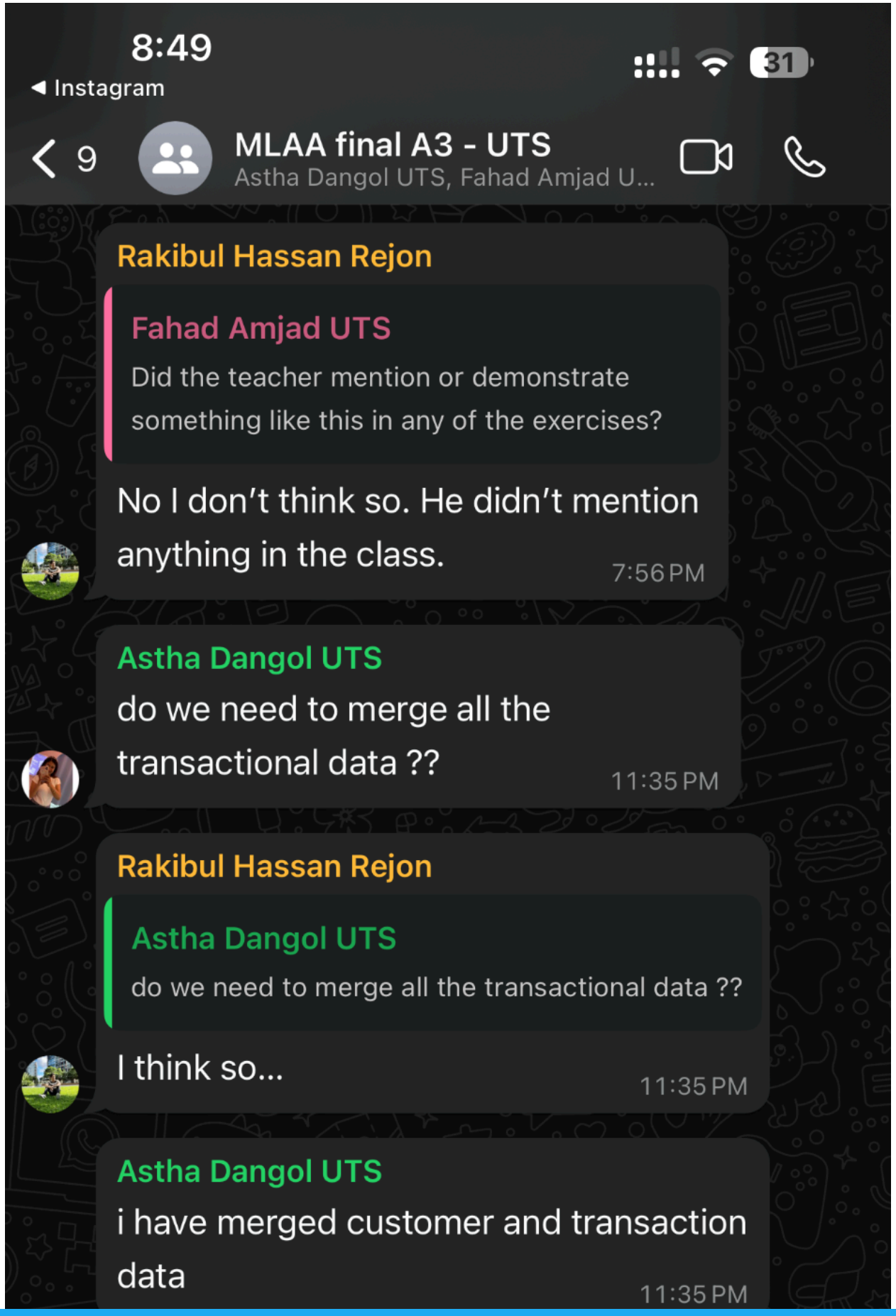

8:48
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Rakibul Hassan Rejon
So, we are going to sit for the meeting at Sunday?
11:11 AM
Yes
11:44 AM
Astha Dangol UTS
i won't be available after 5pm on sunday
2:19 PM
can we do it in the morning?
2:20 PM
Rakibul Hassan Rejon
Astha Dangol UTS
can we do it in the morning?
I am okay with morning as well...
2:20 PM
Fahad Amjad UTS
Ok let's do morning around 9am
8:03 PM
On Sunday

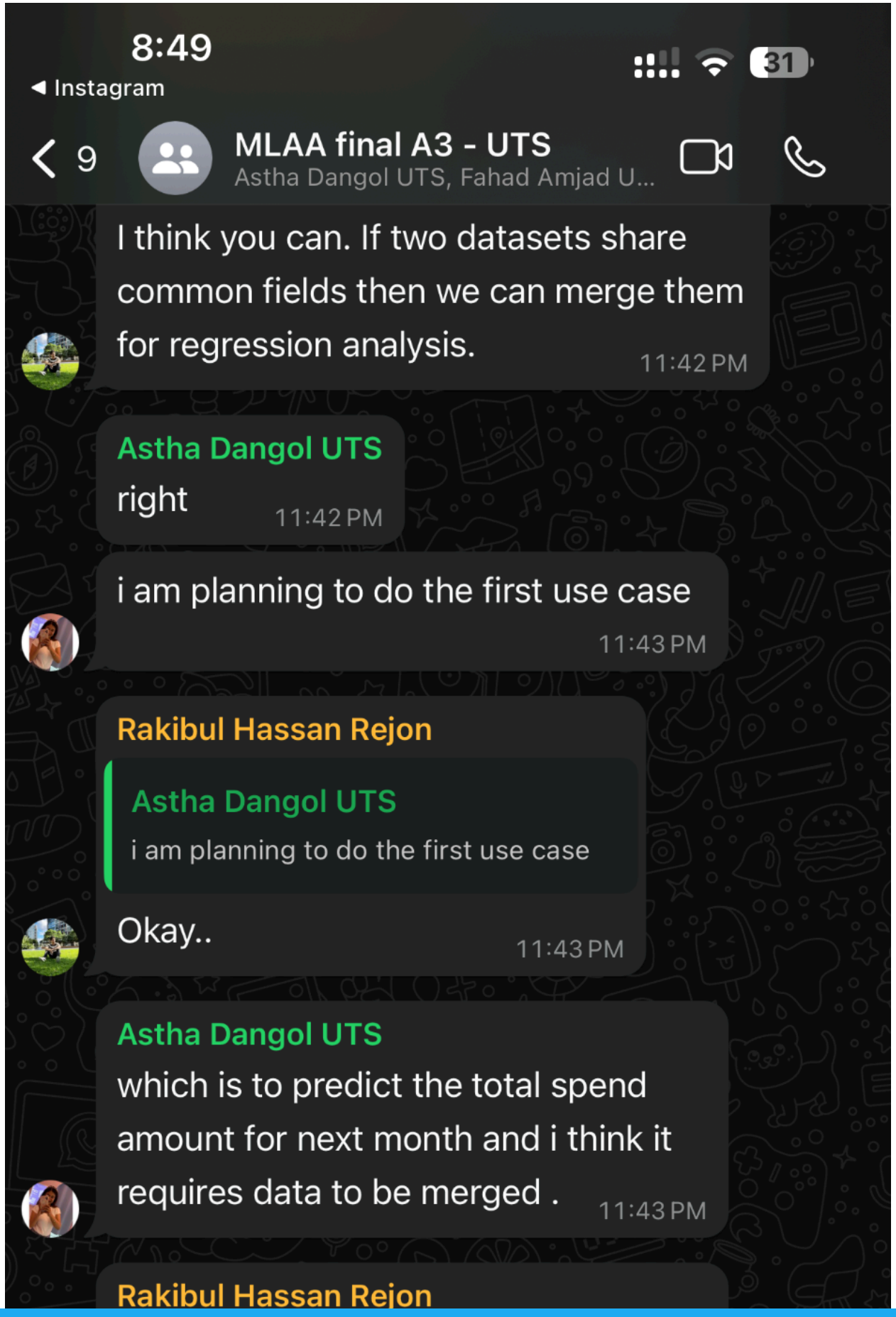

8:48
Instagram
31
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Rakibul Hassan Rejon
36106 Machine Learning Algorithms and Applications - Aut...
Reply
Replies
Anthony So
9 May 2024 at 11:48 AM
Hi Ashley,
Each student of the group has to work on a single use case. So if you are a group of 4, there will be 4 different uses cases but only 1 assigned to each student. Each student has to submit a notebook with the final (best approach). No need to include code for failed experiments. So each student has to submit his own notebook for his/her use case.
Each group has to submit 1 final report. This report is common to all the team members and all use cases.
There will be only a single submission per group. So someone in the group has to gather the different notebooks and final report and submit them on Canvas on behalf of the group. Make sure to clearly define who that person will be and what are the
3:51 PM
Fahad Amjad UTS
The use cases examples have been highlighted in lecture slides for week 6 or 7 - I was going through them last

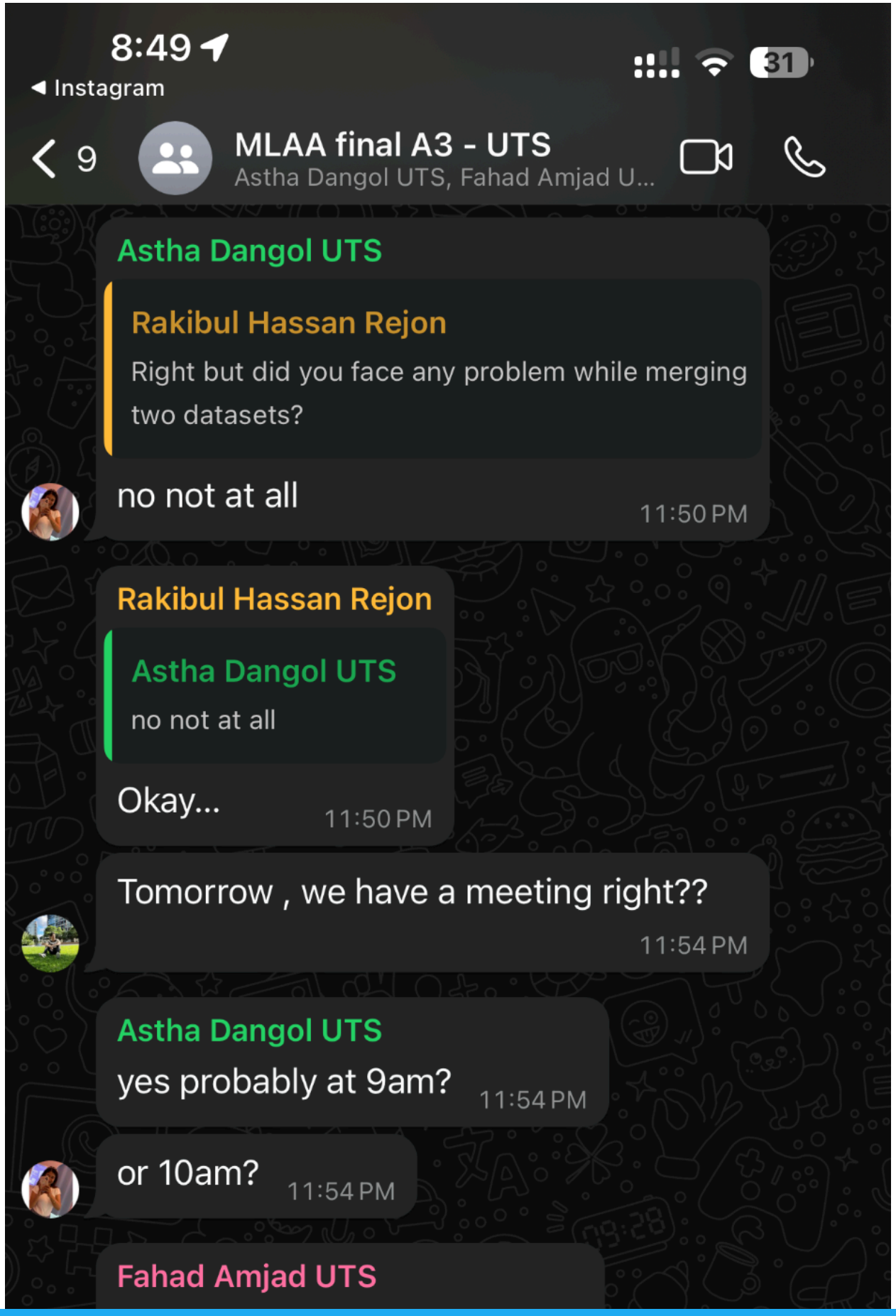

8:48
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
36106 Machine Learning Algorithms and Applications - Autumn 20...
Classification analysis
Unsupervised learning such as clustering, anomaly detection, etc
So each student has to define a business use case, translate it into a data mining problem and then build Machine Learning solutions accordingly. Group members need to collaborate in order to assign different business use cases, share code and insights.
Here are some examples of potential use cases:
Helping customers to better budget their finances by predicting their total spending amount for the next month (regression or classification)
Helping Compliance Team to identify fraudulent behavior by predicting if a transaction is a fraud or not (classification)
Helping Sales Team to target customers most likely to have a significant increase in spending in the next 3 months (regression or classification)
Helping Marketing Team to send customized marketing emails to groups of customers presenting similar spending behaviors (clustering)
Helping Customer Support team to reach out to customers with abnormal behaviors from their usual spending patterns (anomaly detection)
7:59 PM
You will also need to fill a notebook containing your final
Dont we follow this?
7:59 PM
Rakibul Hassan Rejon
Yeah we can follow these questions but these are only examples. If we want we

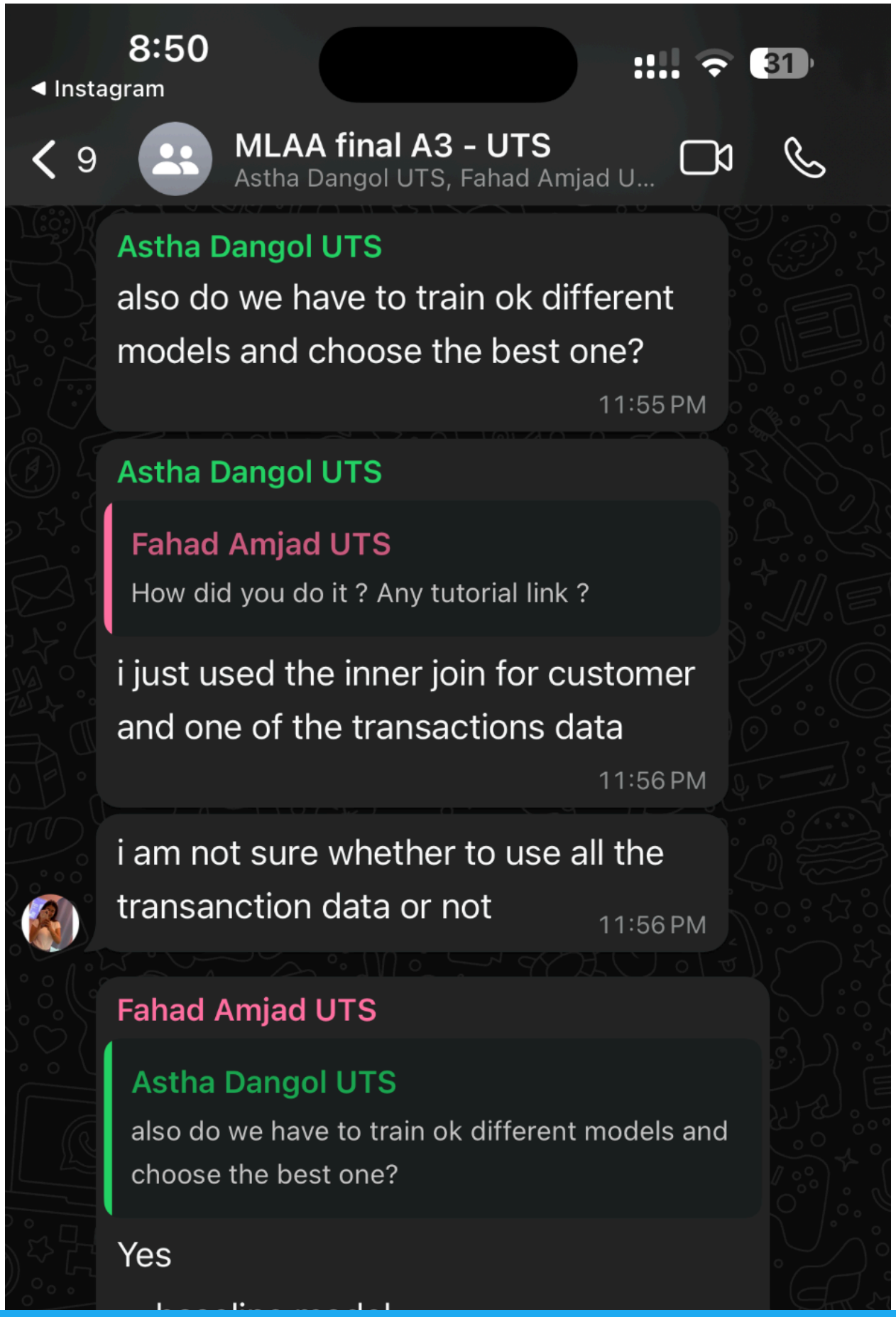

8:49
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Cause i have not yet explored anything of this assignment. Stuck with the other two
8:03 PM
Rakibul Hassan Rejon
You
We should discuss this on our meet
Okay .
8:03 PM
Sat, 11 May
Fahad Amjad UTS
I was looking at the dataset folder - it has several excel files and the data is split in many files and the customer information is in a separate excel sheet - are we supposed to merge all the excel sheets and then utilise the models to get prediction results ?
12:39 AM

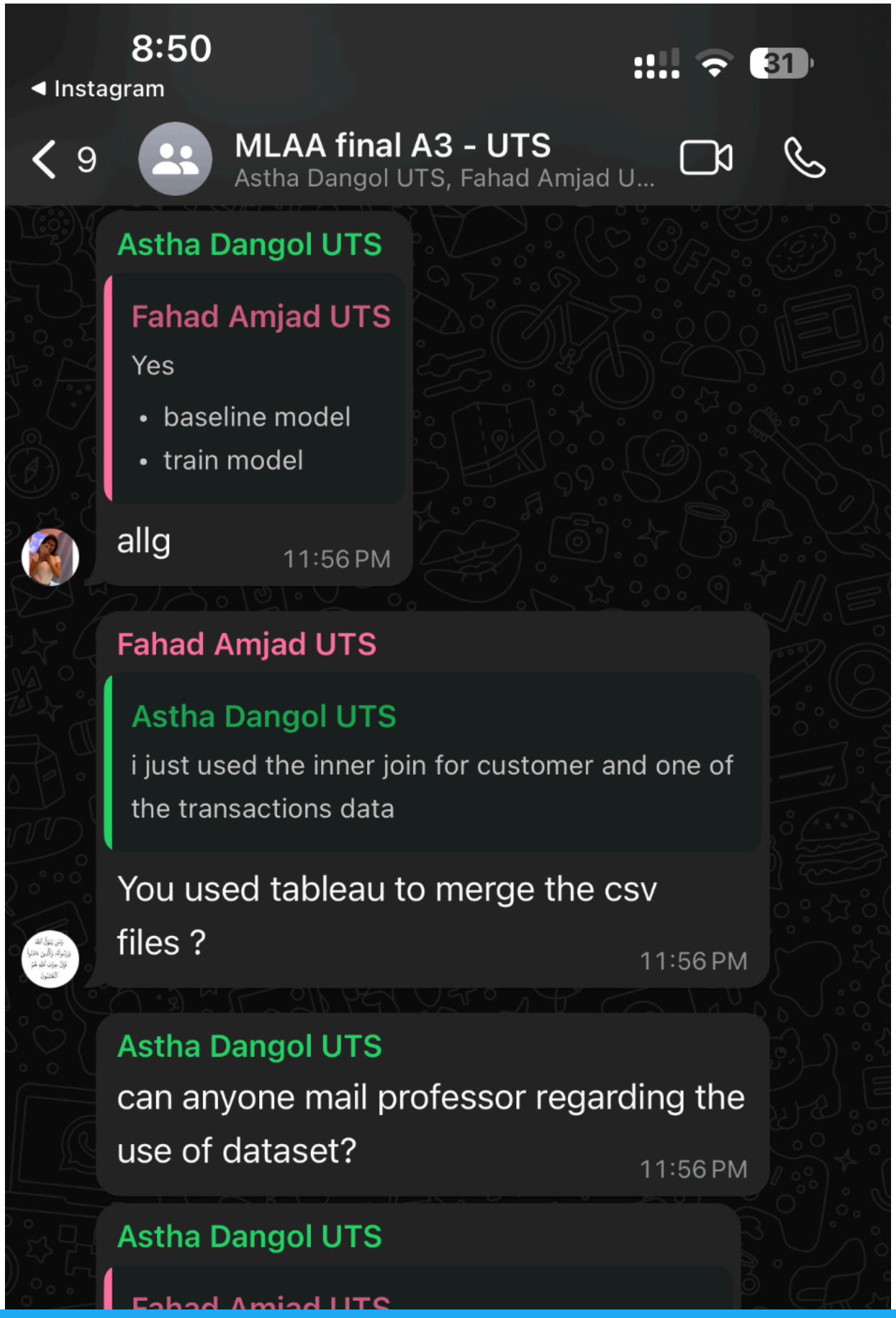

8:49
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Fahad Amjad UTS
I was looking at the dataset folder - it has several excel files and the data is split in many files and the customer information is in a separate excel...
same question
7:27 AM
Rakibul Hassan Rejon
Astha Dangol UTS
same question
I think we need to work with two different excels
7:31 PM
One excel sheet is going to cover all the transactional data and other one is the customer data excel.
7:32 PM
Fahad Amjad UTS
Rakibul Hassan Rejon
One excel sheet is going to cover all the transactional data and other one is the customer

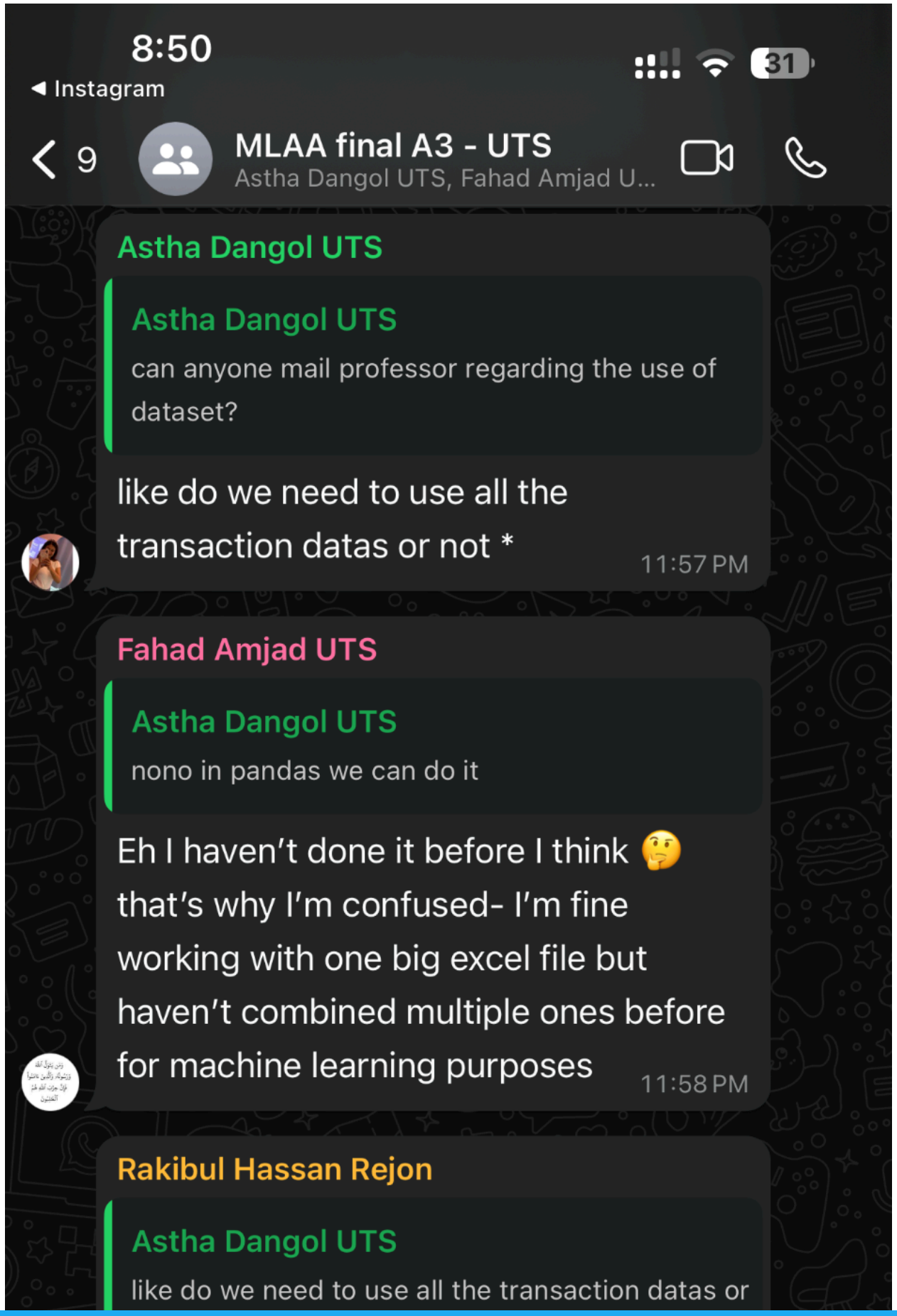

8:49
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Rakibul Hassan Rejon
Fahad Amjad UTS
Did the teacher mention or demonstrate something like this in any of the exercises?
No I don't think so. He didn't mention anything in the class.
7:56 PM
Astha Dangol UTS
do we need to merge all the transactional data ??
11:35 PM
Rakibul Hassan Rejon
Astha Dangol UTS
do we need to merge all the transactional data ??
I think so...
11:35 PM
Astha Dangol UTS
i have merged customer and transaction data
11:35 PM


8:49
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
I think you can. If two datasets share common fields then we can merge them for regression analysis.
11:42 PM
Astha Dangol UTS
right
11:42 PM
i am planning to do the first use case
11:43 PM
Rakibul Hassan Rejon
Astha Dangol UTS
i am planning to do the first use case
Okay..
11:43 PM
Astha Dangol UTS
which is to predict the total spend amount for next month and i think it requires data to be merged .
11:43 PM
Rakibul Hassan Rejon


8:49
Instagram
31
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Astha Dangol UTS
Rakibul Hassan Rejon
Right but did you face any problem while merging two datasets?
no not at all
11:50 PM
Rakibul Hassan Rejon
Astha Dangol UTS
no not at all
Okay...
11:50 PM
Tomorrow , we have a meeting right??
11:54 PM
Astha Dangol UTS
yes probably at 9am?
11:54 PM
or 10am?
11:54 PM
Fahad Amjad UTS


8:50
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Astha Dangol UTS
also do we have to train ok different models and choose the best one?
11:55 PM
Astha Dangol UTS
Fahad Amjad UTS
How did you do it ? Any tutorial link ?
i just used the inner join for customer and one of the transactions data
11:56 PM
i am not sure whether to use all the transanction data or not
11:56 PM
Fahad Amjad UTS
Astha Dangol UTS
also do we have to train ok different models and choose the best one?
Yes


8:50
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Astha Dangol UTS
Fahad Amjad UTS
Yes
baseline model
train model
allg
11:56 PM
Fahad Amjad UTS
Astha Dangol UTS
i just used the inner join for customer and one of the transactions data
You used tableau to merge the csv files ?
11:56 PM
Astha Dangol UTS
can anyone mail professor regarding the use of dataset?
11:56 PM
Astha Dangol UTS


8:50
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Astha Dangol UTS
Astha Dangol UTS
can anyone mail professor regarding the use of dataset?
like do we need to use all the transaction datas or not *
11:57 PM
Fahad Amjad UTS
Astha Dangol UTS
nono in pandas we can do it
Eh I haven't done it before I think 🤔 that's why I'm confused- I'm fine working with one big excel file but haven't combined multiple ones before for machine learning purposes
11:58 PM
Rakibul Hassan Rejon
Astha Dangol UTS
like do we need to use all the transaction datas or

8:50

◄ Instagram

9

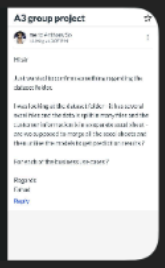
MLAA final A3 - UTS

Astha Dangol UTS, Fahad Amjad U...

Astha Dangol UTS

Fahad Amjad UTS

Already asked him waiting for response



oh that's great 11:59 PM

Fahad Amjad UTS

36106 Machine Learning Algorithms and Applicatio...

Huge Datasets for AT3

TL **Tyson Lee**
9 May 2024 at 09:44

Hey Anthony, hope you are well!

Just wondering what best-practice is when we have a significant amount of data - meaning it will make a long time for our models to be trained make a prediction. Can we select say 20% of the dataset for training, testing and validation? Is there a rule we should follow?

Tyson

Reply

8:50

◄ Instagram

9

MLAA final A3 - UTS

Astha Dangol UTS, Fahad Amjad U...

Sun, 12 May

Astha Dangol UTS

> **Fahad Amjad UTS**
> Photo

does that mean we can go ahead with transaction0 data only 12:00 AM

i think the shape of the data in transaction_0 should be fine then 12:00 AM

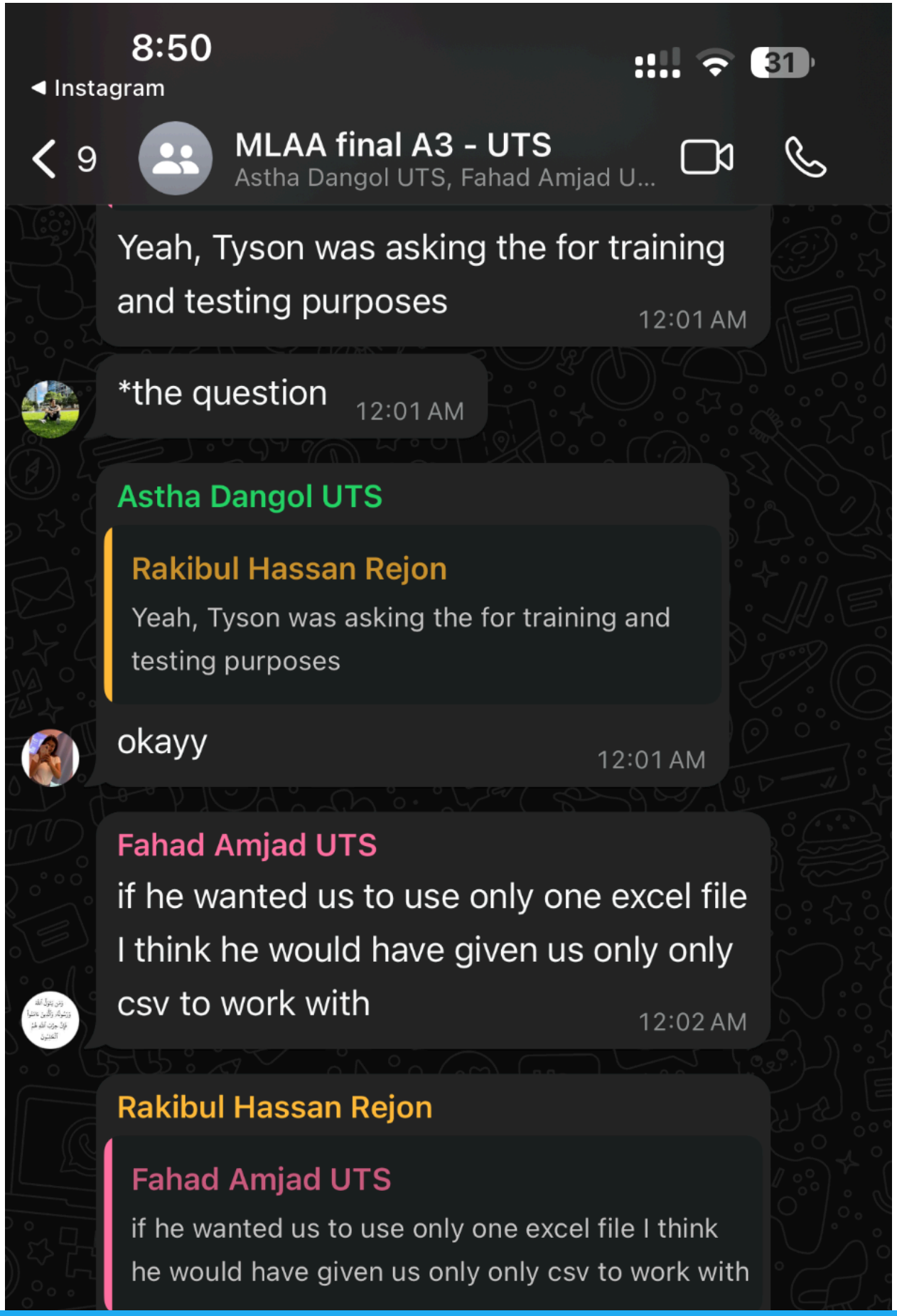
Fahad Amjad UTS

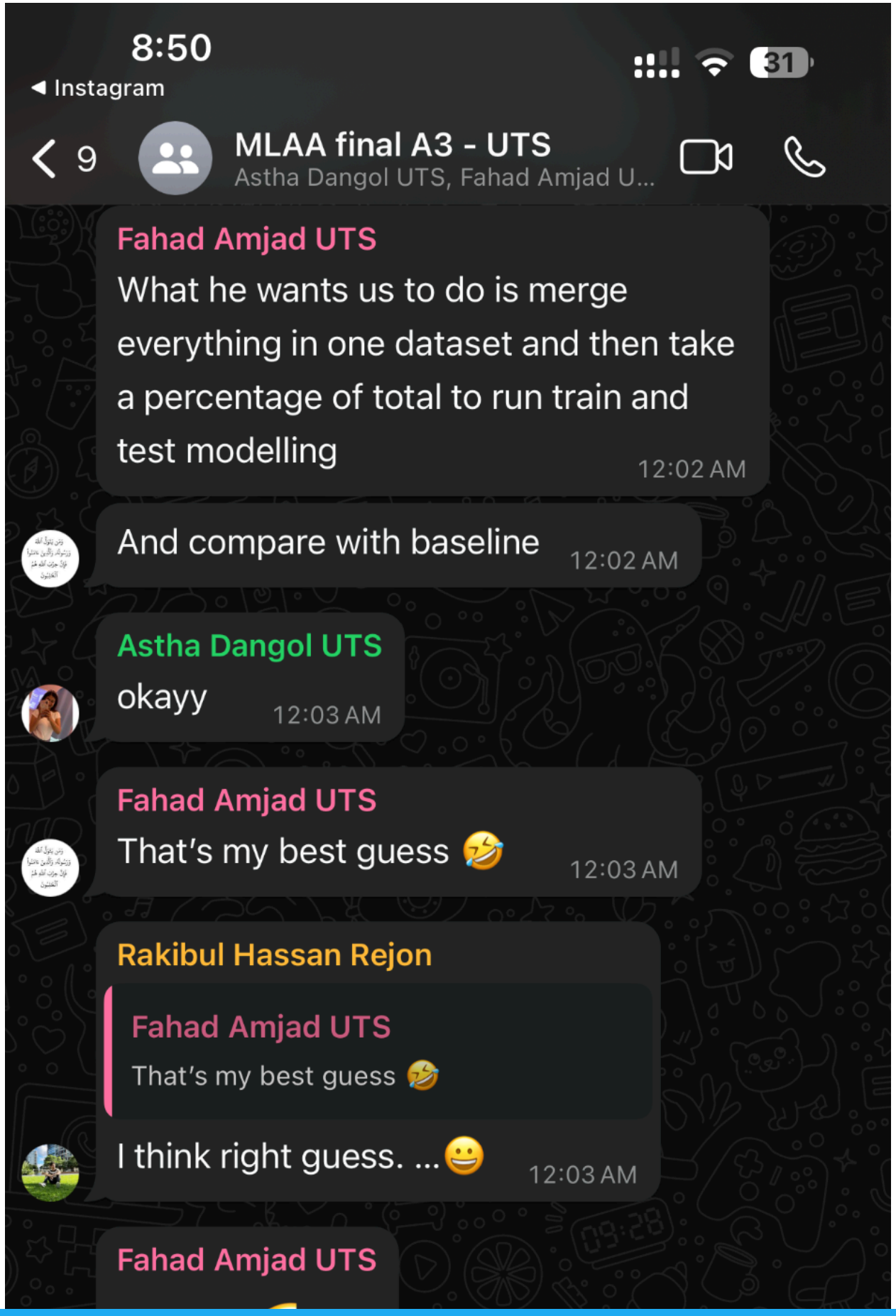
No I think we have to combine all and take a percentage of total dataset for training and testing I think 12:00 AM

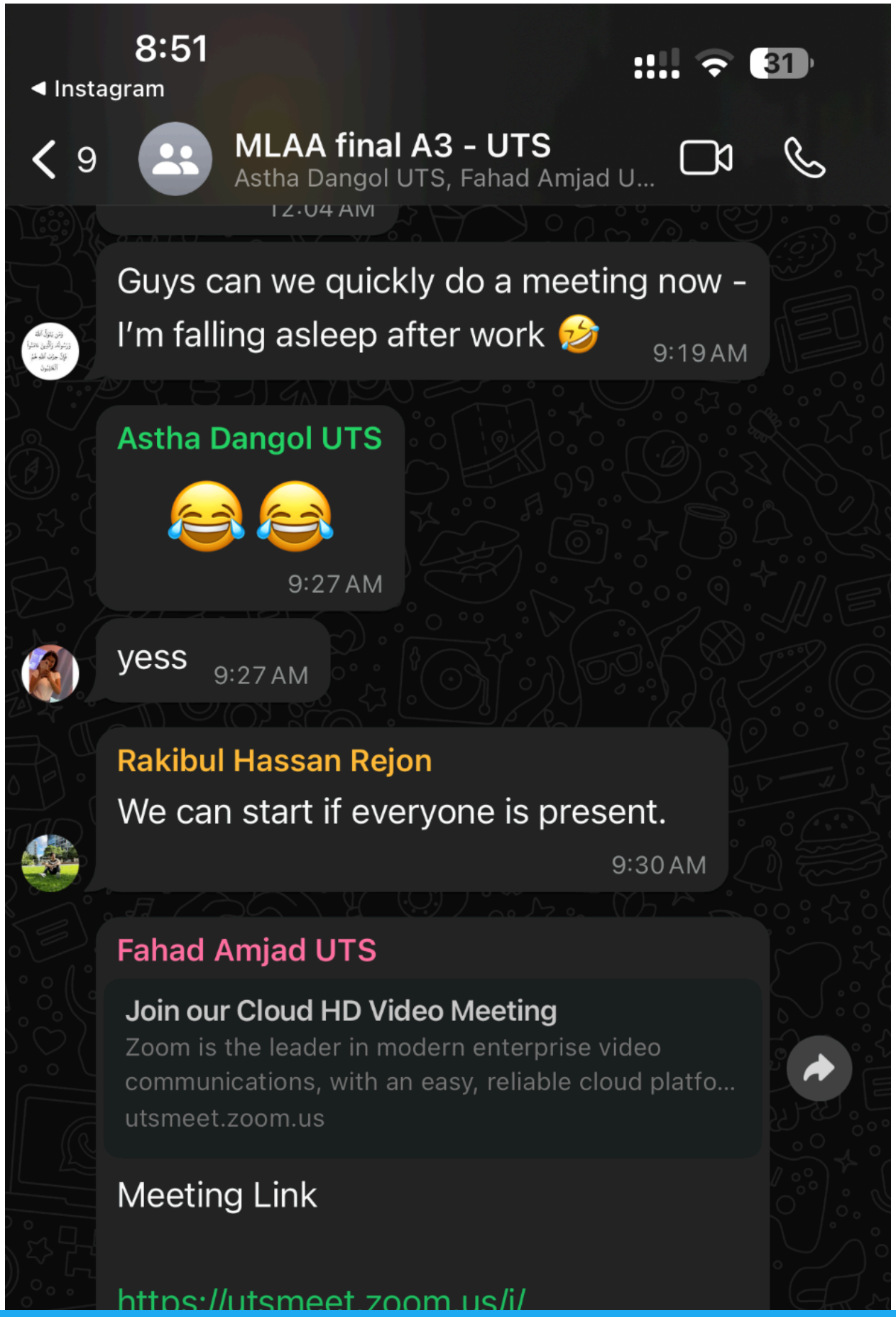
Otherwise the results will not reflect data from overall files 12:01 AM

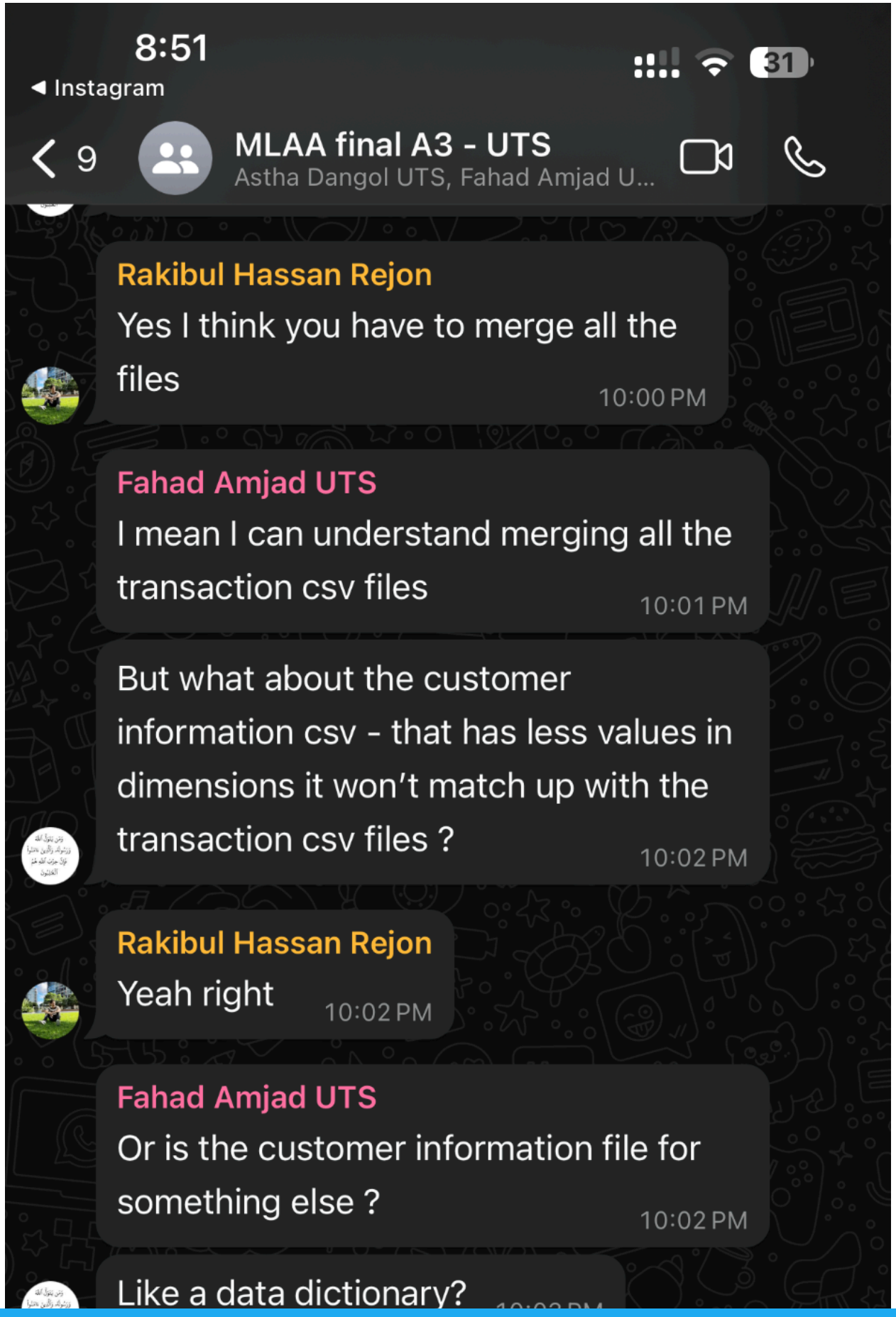
Astha Dangol UTS

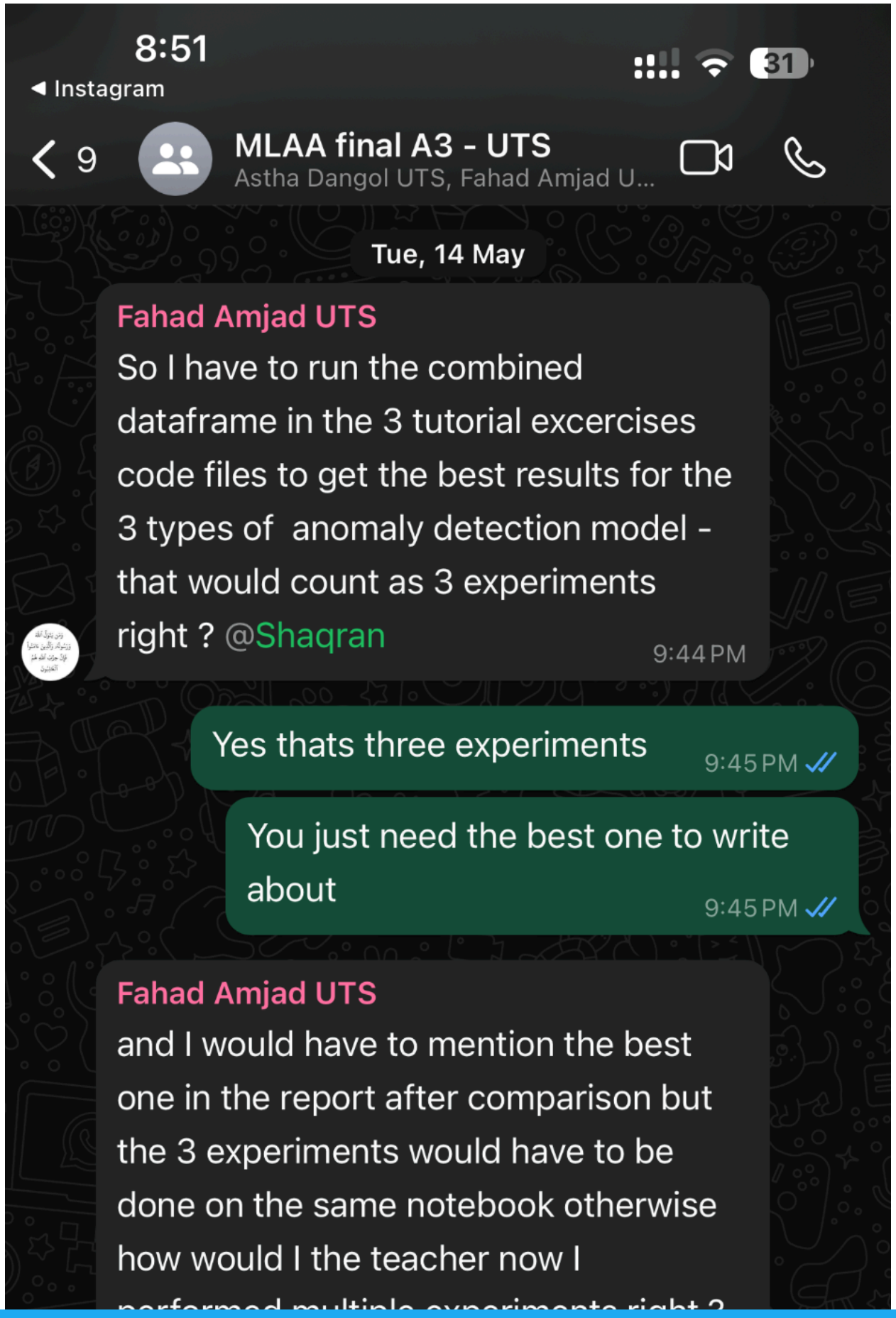
that is true

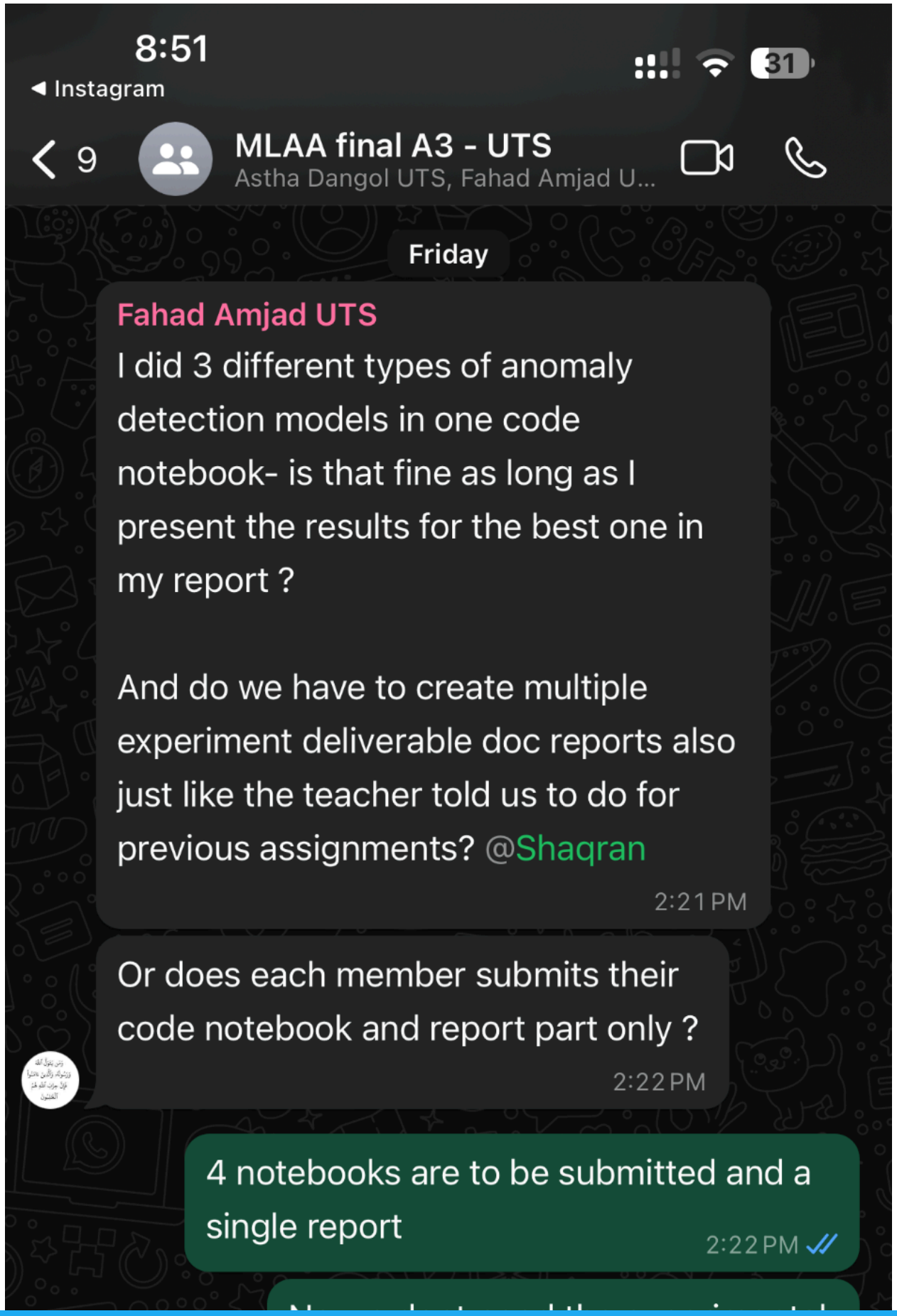

8:50
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Yeah, Tyson was asking the for training and testing purposes
12:01 AM
*the question
12:01 AM
Astha Dangol UTS
Rakibul Hassan Rejon
Yeah, Tyson was asking the for training and testing purposes
okayy
12:01 AM
Fahad Amjad UTS
if he wanted us to use only one excel file I think he would have given us only only csv to work with
12:02 AM
Rakibul Hassan Rejon
Fahad Amjad UTS
if he wanted us to use only one excel file I think he would have given us only only csv to work with

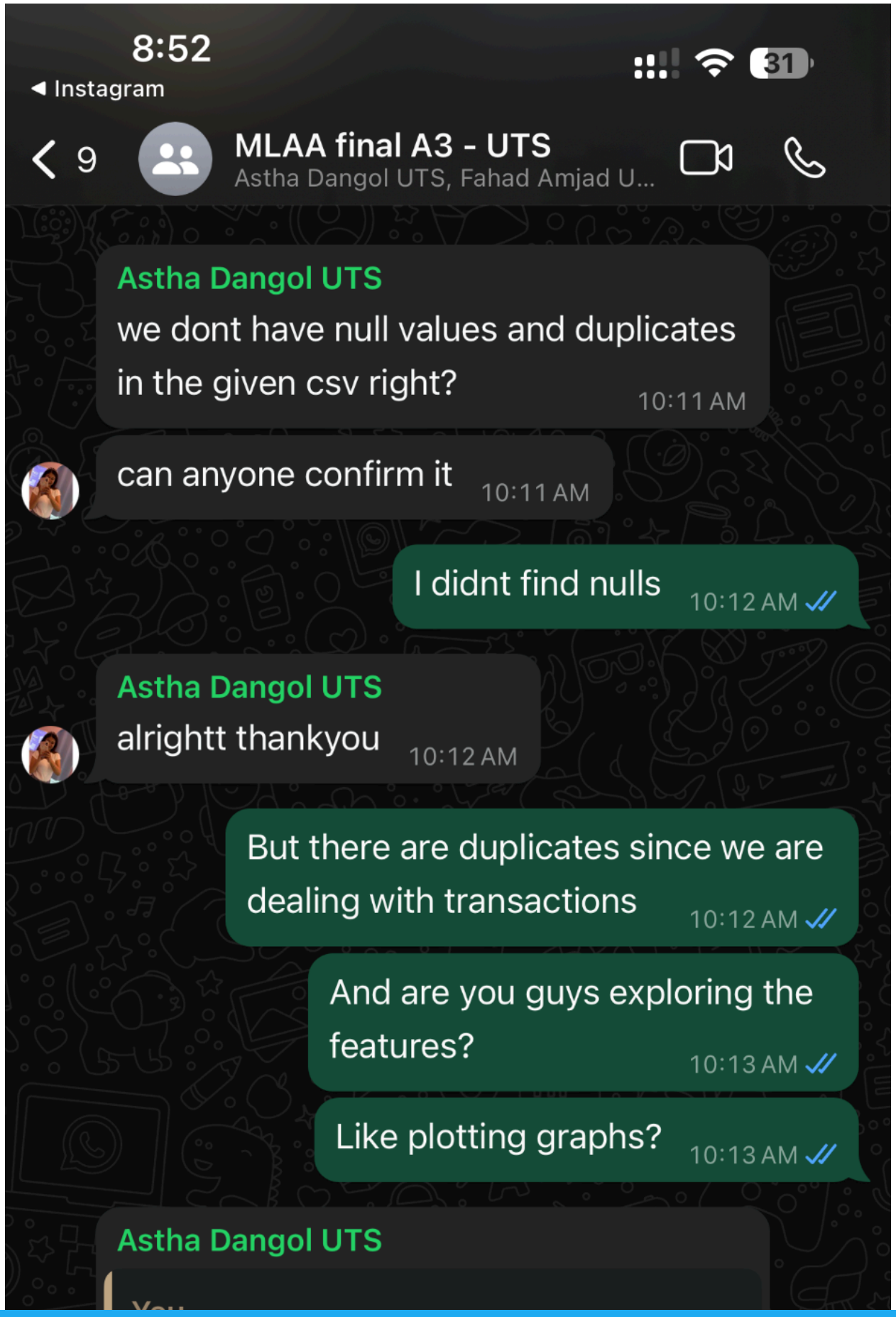

8:50
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Fahad Amjad UTS
What he wants us to do is merge everything in one dataset and then take a percentage of total to run train and test modelling
12:02 AM
And compare with baseline
12:02 AM
Astha Dangol UTS
okayy
12:03 AM
Fahad Amjad UTS
That's my best guess
12:03 AM
Rakibul Hassan Rejon
Fahad Amjad UTS
That's my best guess
I think right guess. ...
12:03 AM
Fahad Amjad UTS

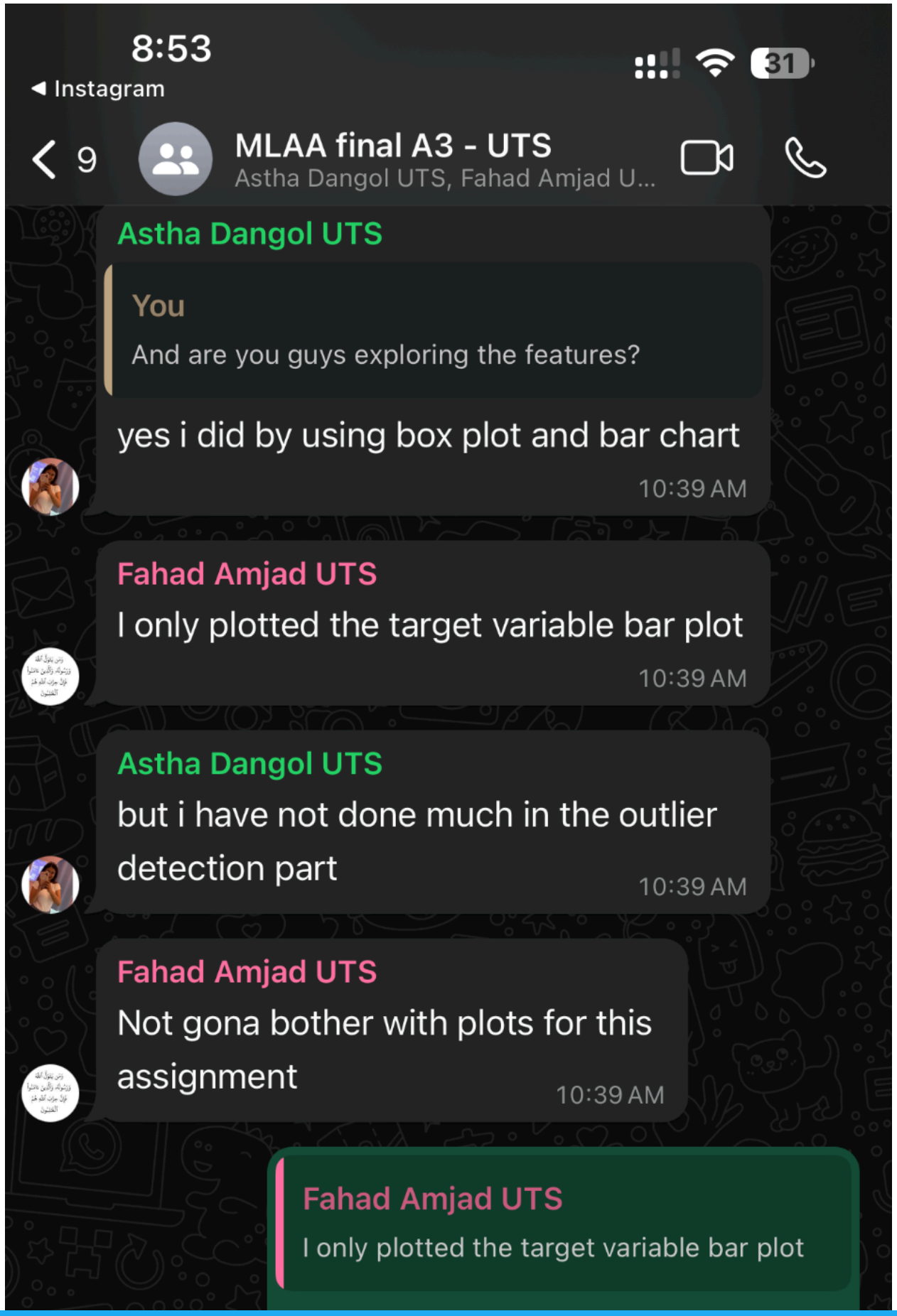

8:51
Instagram
31
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
12:04 AM
Guys can we quickly do a meeting now - I'm falling asleep after work 🤣
9:19 AM
Astha Dangol UTS
😂😂
9:27 AM
yess
9:27 AM
Rakibul Hassan Rejon
We can start if everyone is present.
9:30 AM
Fahad Amjad UTS
Join our Cloud HD Video Meeting
Zoom is the leader in modern enterprise video communications, with an easy, reliable cloud platfo...
utsmeet.zoom.us
Meeting Link
https://utsmeet.zoom.us/j/

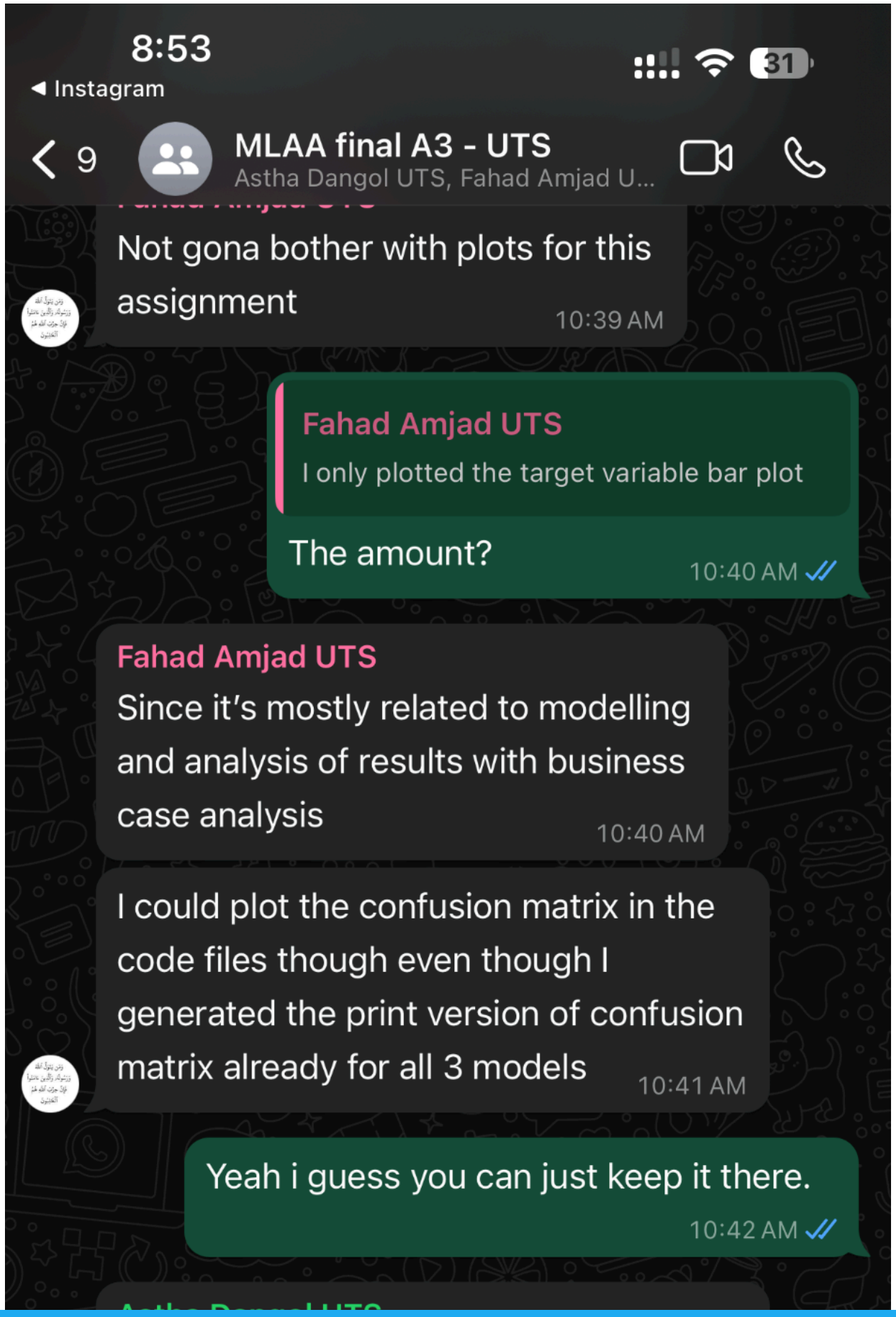

8:51
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Rakibul Hassan Rejon
Yes I think you have to merge all the files
10:00 PM
Fahad Amjad UTS
I mean I can understand merging all the transaction csv files
10:01 PM
But what about the customer information csv - that has less values in dimensions it won't match up with the transaction csv files ?
10:02 PM
Rakibul Hassan Rejon
Yeah right
10:02 PM
Fahad Amjad UTS
Or is the customer information file for something else ?
10:02 PM
Like a data dictionary?

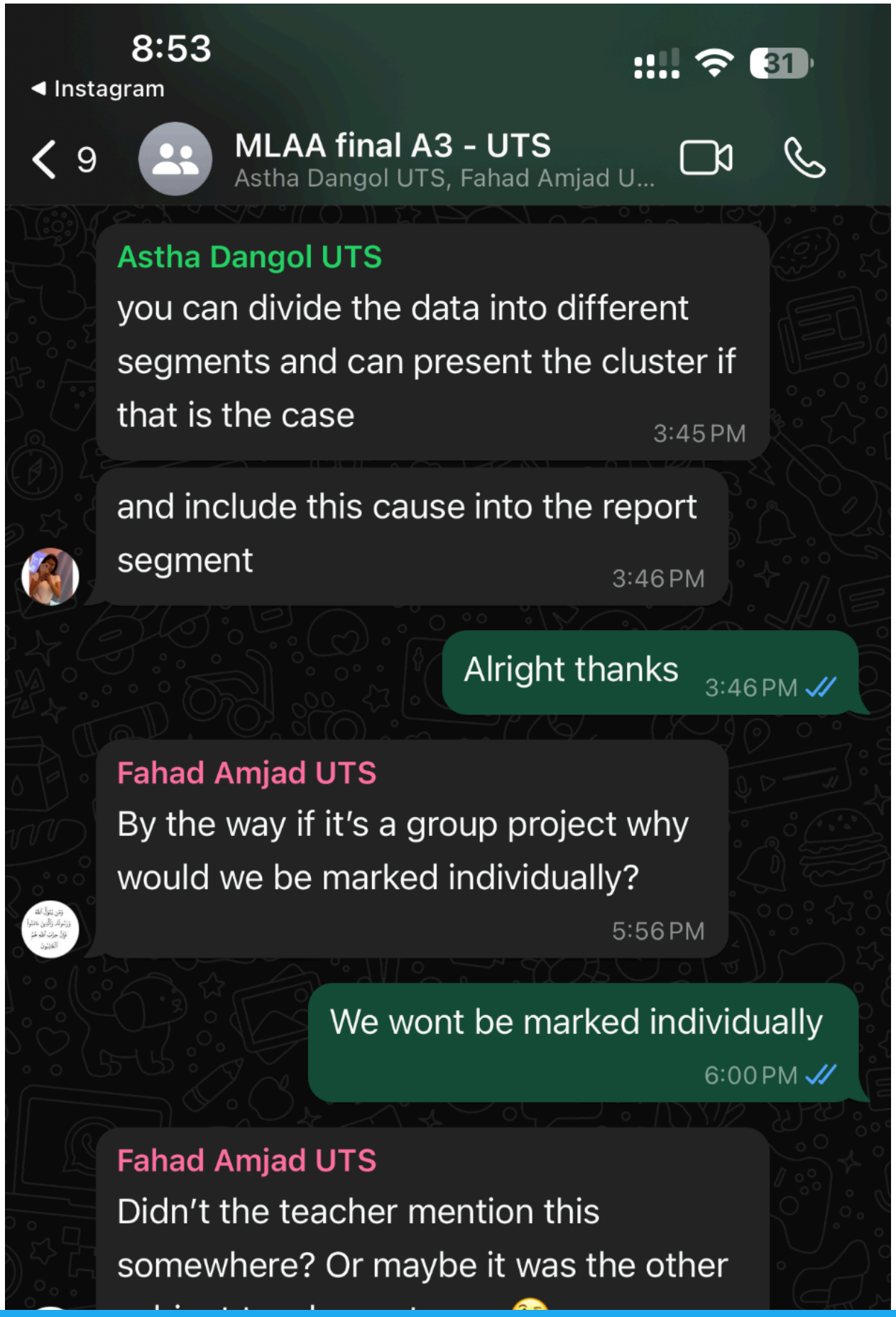

8:51
Instagram
31
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Tue, 14 May
Fahad Amjad UTS
So I have to run the combined dataframe in the 3 tutorial excercises code files to get the best results for the 3 types of anomaly detection model - that would count as 3 experiments right ? @Shaqran
9:44 PM
Yes thats three experiments
9:45 PM
You just need the best one to write about
9:45 PM
Fahad Amjad UTS
and I would have to mention the best one in the report after comparison but the 3 experiments would have to be done on the same notebook otherwise how would I the teacher now I

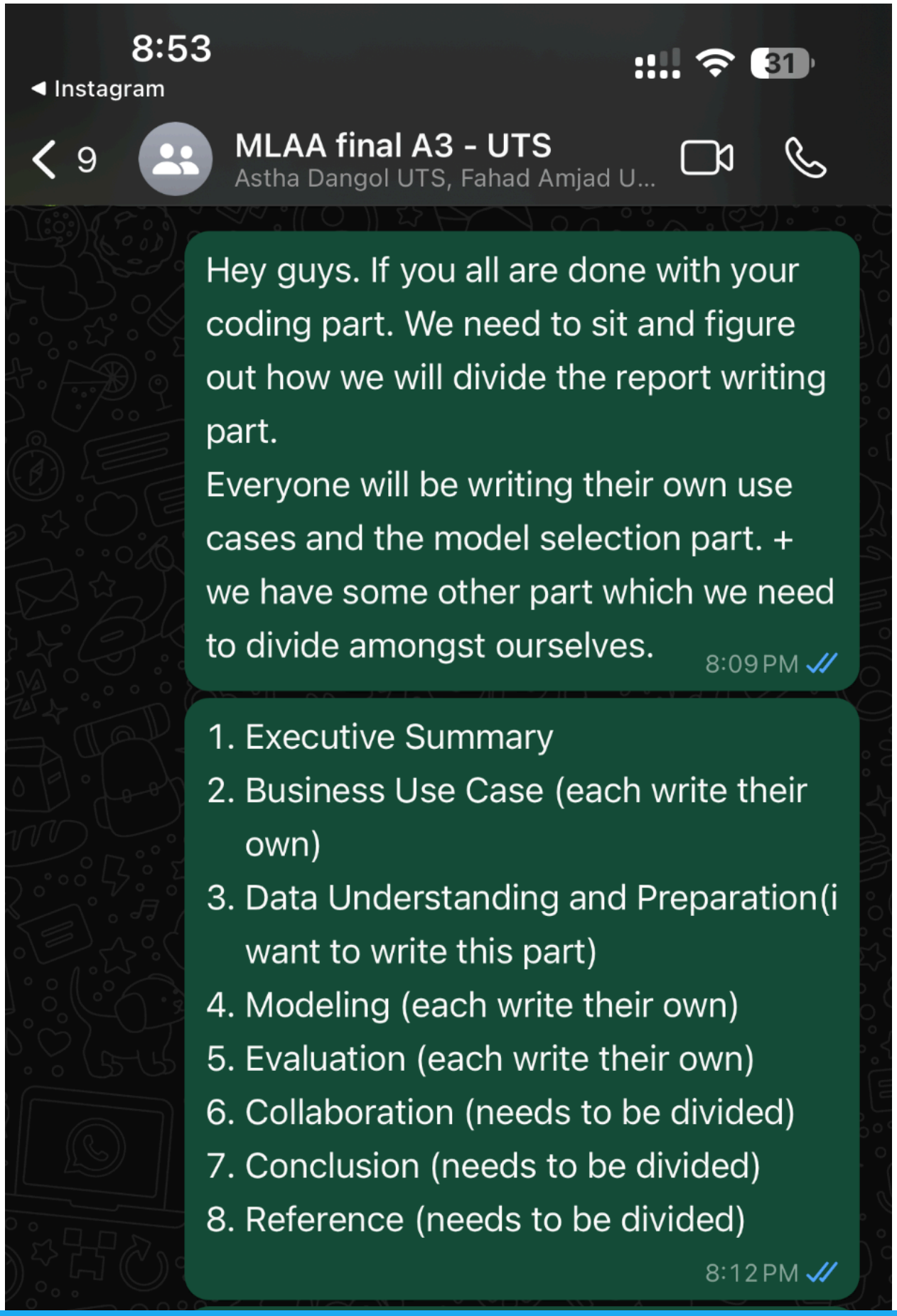

8:51
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Friday
Fahad Amjad UTS
I did 3 different types of anomaly detection models in one code notebook- is that fine as long as I present the results for the best one in my report ?
And do we have to create multiple experiment deliverable doc reports also just like the teacher told us to do for previous assignments? @Shaqran
2:21 PM
Or does each member submits their code notebook and report part only ?
2:22 PM
4 notebooks are to be submitted and a single report
2:22 PM

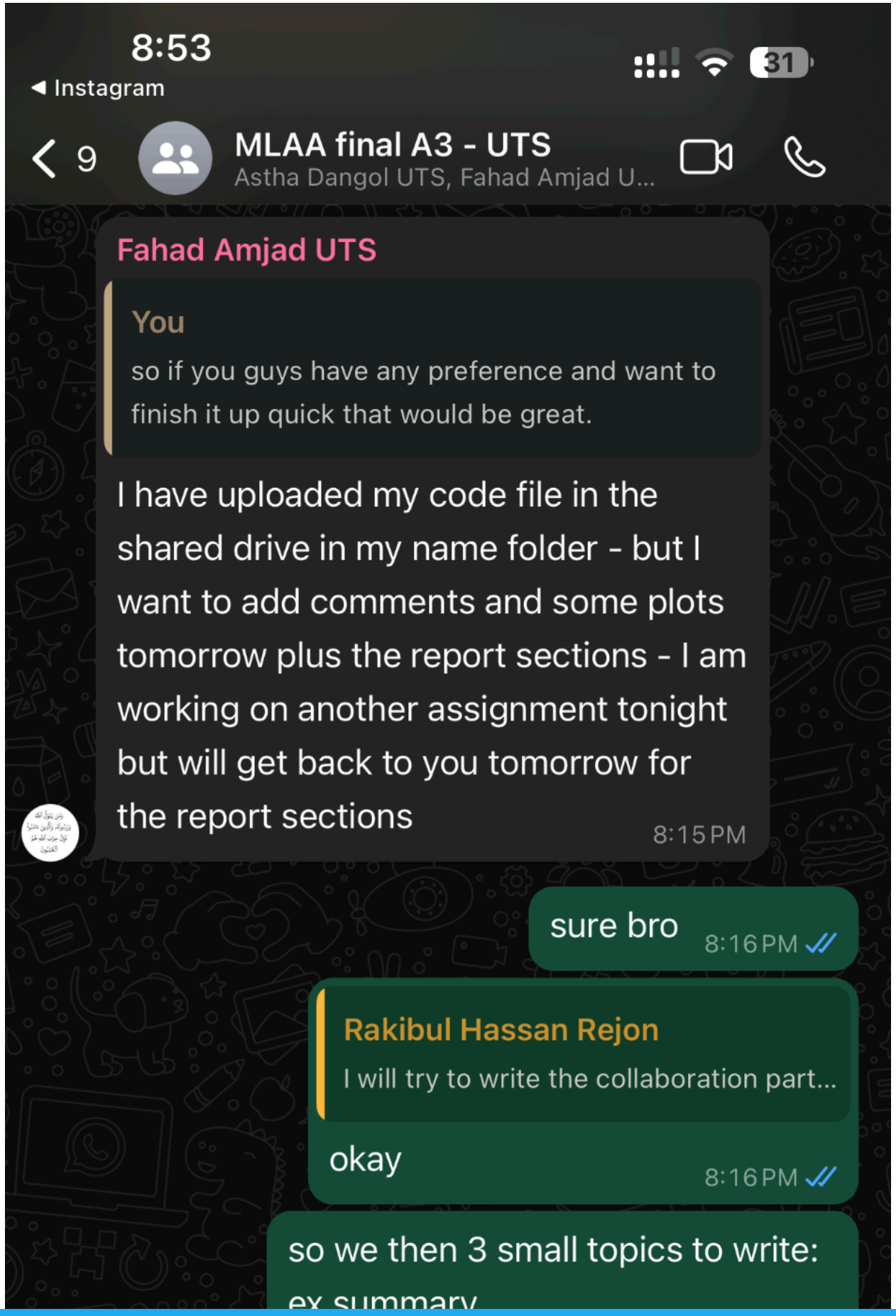

8:52
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Astha Dangol UTS
we dont have null values and duplicates in the given csv right?
10:11 AM
can anyone confirm it
10:11 AM
I didnt find nulls
10:12 AM
Astha Dangol UTS
alrightt thankyou
10:12 AM
But there are duplicates since we are dealing with transactions
10:12 AM
And are you guys exploring the features?
10:13 AM
Like plotting graphs?
10:13 AM
Astha Dangol UTS

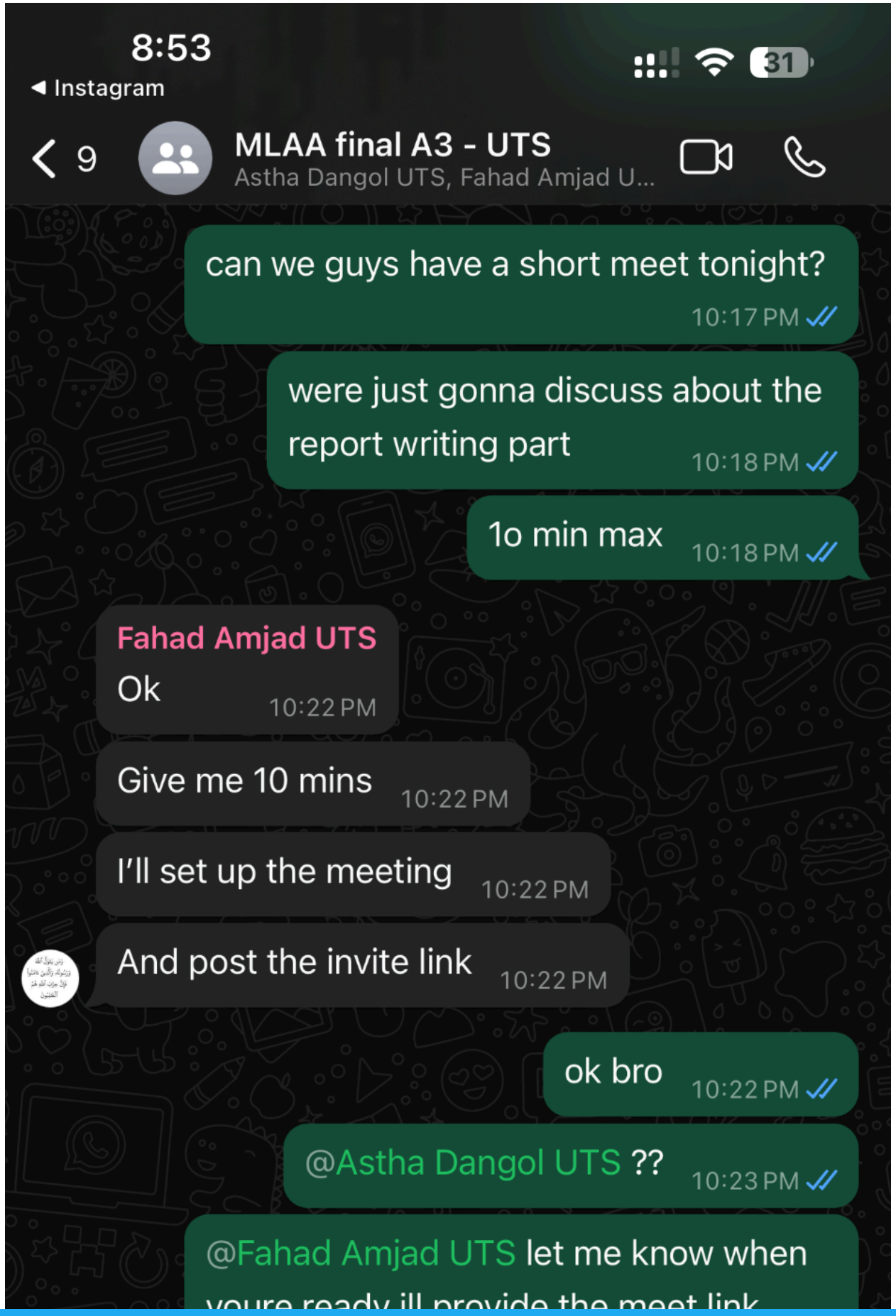

8:53
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Astha Dangol UTS
You
And are you guys exploring the features?
yes i did by using box plot and bar chart
10:39 AM
Fahad Amjad UTS
I only plotted the target variable bar plot
10:39 AM
Astha Dangol UTS
but i have not done much in the outlier detection part
10:39 AM
Fahad Amjad UTS
Not gona bother with plots for this assignment
10:39 AM
Fahad Amjad UTS
I only plotted the target variable bar plot

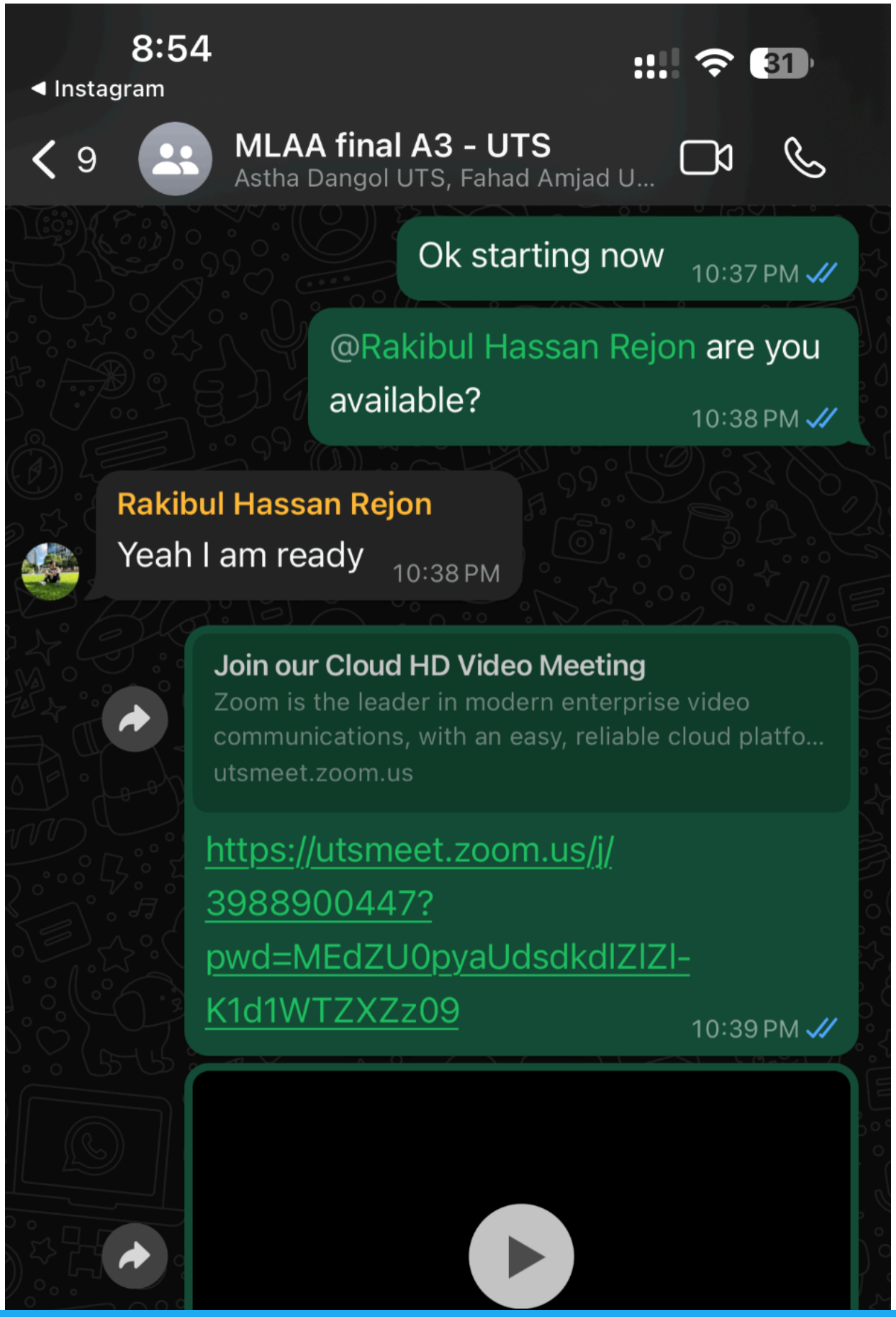

8:53
Instagram
31
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Not gona bother with plots for this assignment
10:39 AM
Fahad Amjad UTS
I only plotted the target variable bar plot
The amount?
10:40 AM
Fahad Amjad UTS
Since it's mostly related to modelling and analysis of results with business case analysis
10:40 AM
I could plot the confusion matrix in the code files though even though I generated the print version of confusion matrix already for all 3 models
10:41 AM
Yeah i guess you can just keep it there.
10:42 AM

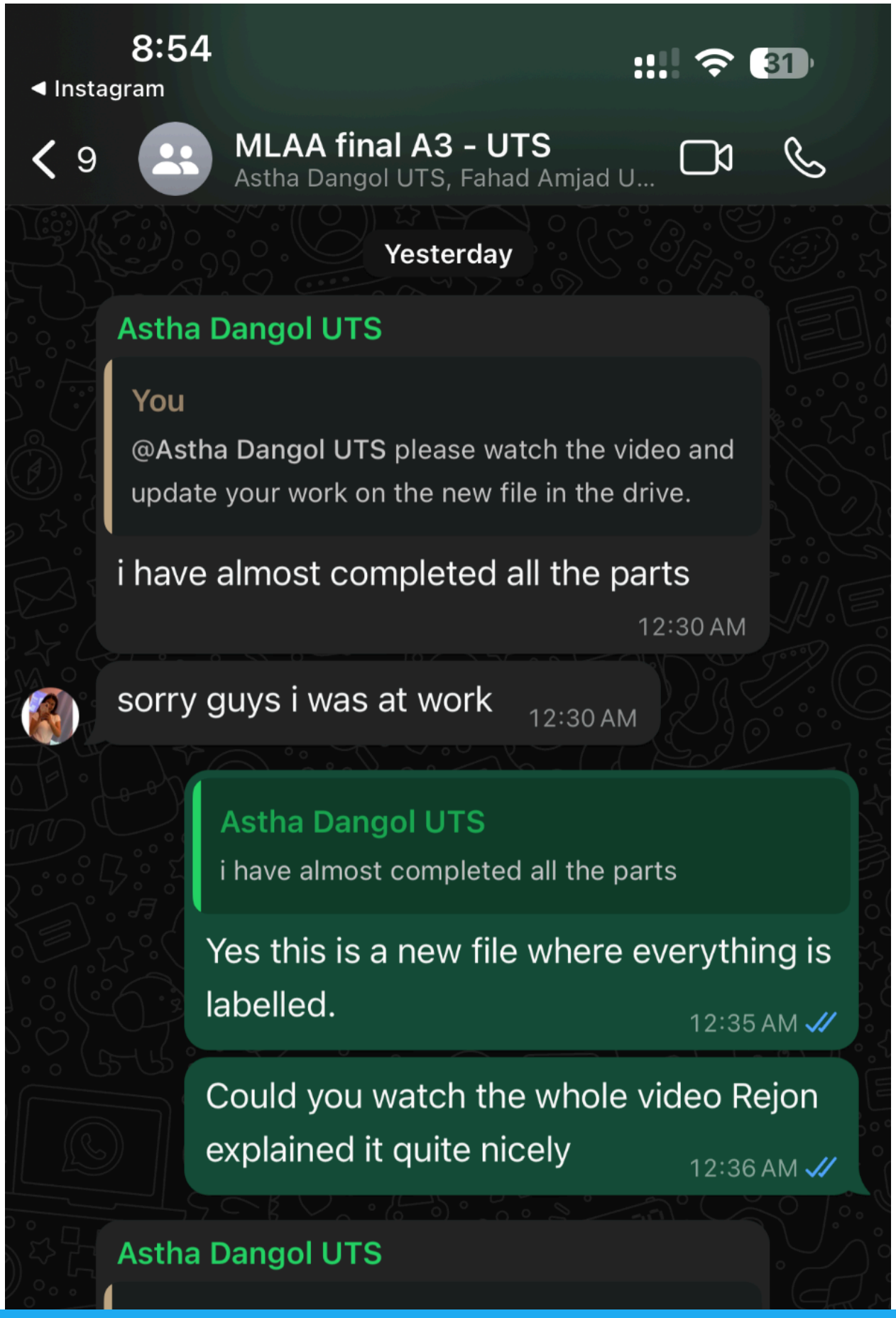

8:53
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Astha Dangol UTS
you can divide the data into different segments and can present the cluster if that is the case
3:45 PM
and include this cause into the report segment
3:46 PM
Alright thanks
3:46 PM
Fahad Amjad UTS
By the way if it's a group project why would we be marked individually?
5:56 PM
We wont be marked individually
6:00 PM
Fahad Amjad UTS
Didn't the teacher mention this somewhere? Or maybe it was the other

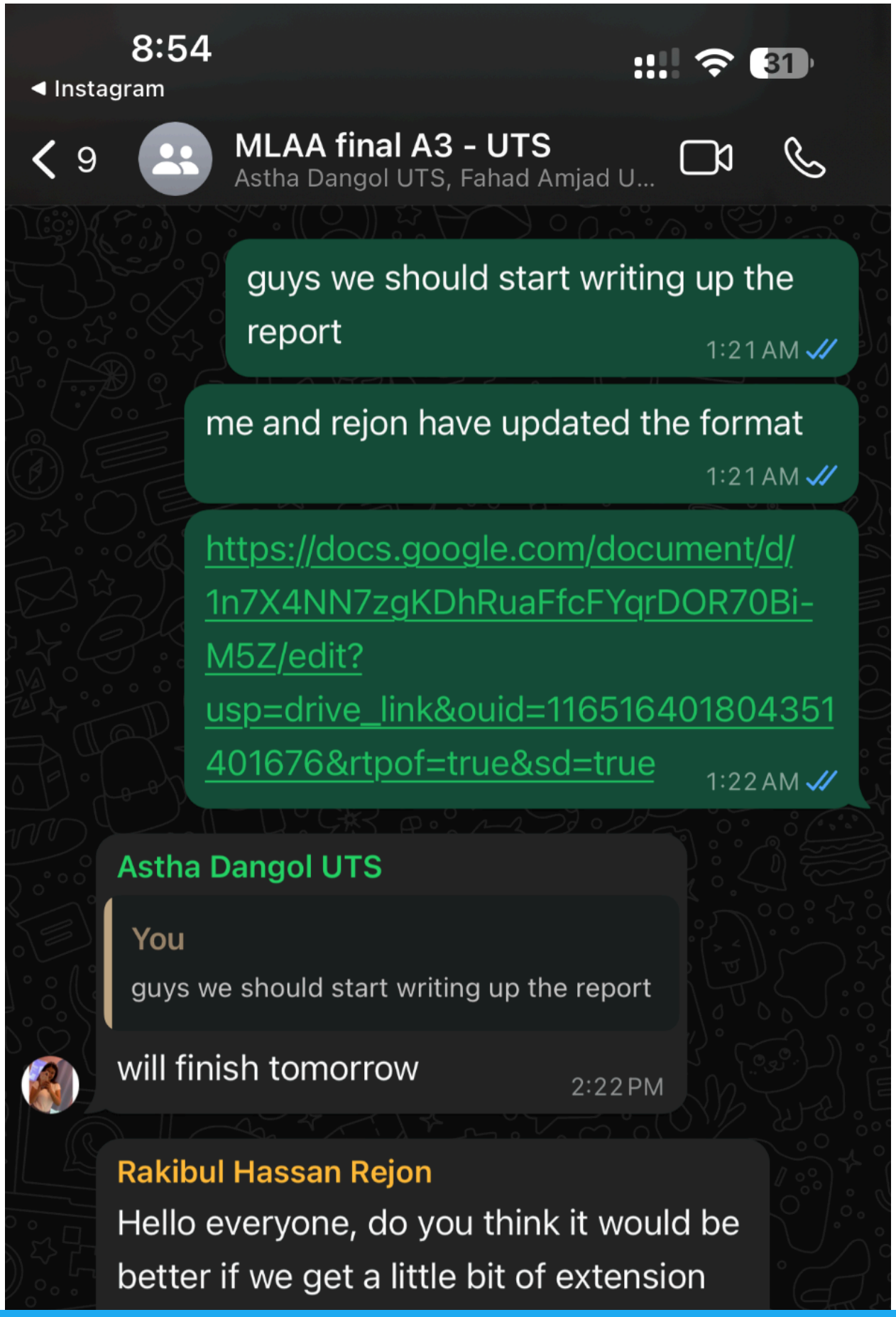

8:53
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Hey guys. If you all are done with your coding part. We need to sit and figure out how we will divide the report writing part.
Everyone will be writing their own use cases and the model selection part. + we have some other part which we need to divide amongst ourselves.
8:09 PM
1. Executive Summary
2. Business Use Case (each write their own)
3. Data Understanding and Preparation(i want to write this part)
4. Modeling (each write their own)
5. Evaluation (each write their own)
6. Collaboration (needs to be divided)
7. Conclusion (needs to be divided)
8. Reference (needs to be divided)
8:12 PM

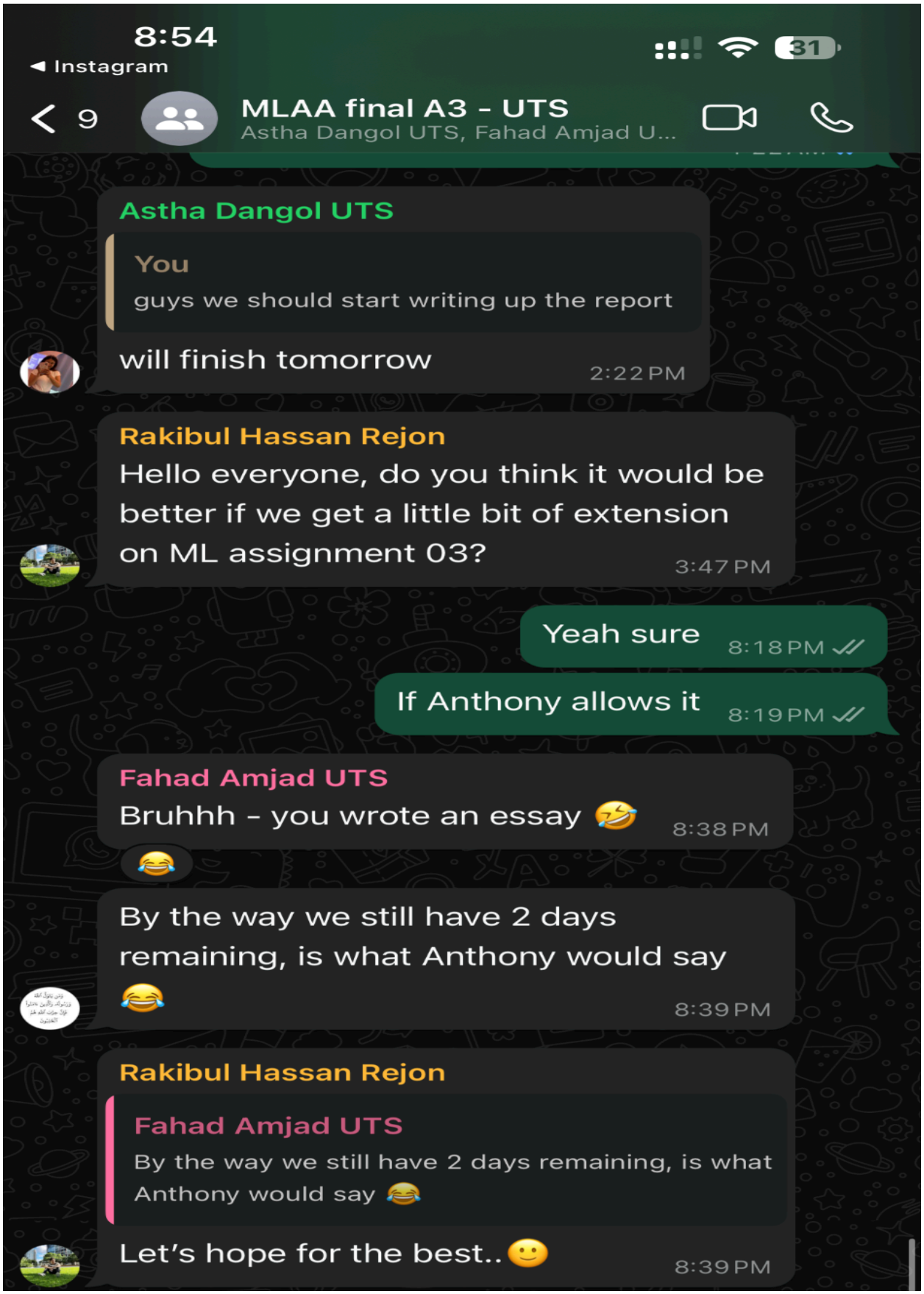

8:53
Instagram
31
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Fahad Amjad UTS
You
so if you guys have any preference and want to finish it up quick that would be great.
I have uploaded my code file in the shared drive in my name folder - but I want to add comments and some plots tomorrow plus the report sections - I am working on another assignment tonight but will get back to you tomorrow for the report sections
8:15 PM
sure bro
8:16 PM
Rakibul Hassan Rejon
I will try to write the collaboration part...
okay
8:16 PM
so we then 3 small topics to write:
ex summary


8:53
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
can we guys have a short meet tonight?
10:17 PM
were just gonna discuss about the report writing part
10:18 PM
1o min max
10:18 PM
Fahad Amjad UTS
Ok
10:22 PM
Give me 10 mins
10:22 PM
I'll set up the meeting
10:22 PM
And post the invite link
10:22 PM
ok bro
10:22 PM
@Astha Dangol UTS ??
10:23 PM
@Fahad Amjad UTS let me know when


8:54
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Ok starting now
10:37 PM
@Rakibul Hassan Rejon are you available?
10:38 PM
Rakibul Hassan Rejon
Yeah I am ready
10:38 PM
Join our Cloud HD Video Meeting
Zoom is the leader in modern enterprise video communications, with an easy, reliable cloud platfo...
utsmeet.zoom.us
https://utsmeet.zoom.us/j/3988900447?pwd=MEdZU0pyaUdsdkdlZlZl-K1d1WTZXZz09
10:39 PM


8:54
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Yesterday
Astha Dangol UTS
You
@Astha Dangol UTS please watch the video and update your work on the new file in the drive.
i have almost completed all the parts
12:30 AM
sorry guys i was at work
12:30 AM
Astha Dangol UTS
i have almost completed all the parts
Yes this is a new file where everything is labelled.
12:35 AM
Could you watch the whole video Rejon explained it quite nicely
12:36 AM
Astha Dangol UTS


8:54
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
guys we should start writing up the report
1:21 AM
me and rejon have updated the format
1:21 AM
https://docs.google.com/document/d/1n7X4NN7zgKDhRuaFfcFYqrDOR70Bi-M5Z/edit?usp=drive_link&ouid=116516401804351401676&rtpof=true&sd=true
1:22 AM
Astha Dangol UTS
You
guys we should start writing up the report
will finish tomorrow
2:22 PM
Rakibul Hassan Rejon
Hello everyone, do you think it would be better if we get a little bit of extension


8:54
Instagram
9
MLAA final A3 - UTS
Astha Dangol UTS, Fahad Amjad U...
Astha Dangol UTS
You
guys we should start writing up the report
will finish tomorrow
2:22 PM
Rakibul Hassan Rejon
Hello everyone, do you think it would be better if we get a little bit of extension on ML assignment 03?
3:47 PM
Yeah sure
8:18 PM
If Anthony allows it
8:19 PM
Fahad Amjad UTS
Bruhhh - you wrote an essay
8:38 PM
By the way we still have 2 days remaining, is what Anthony would say
8:39 PM
Rakibul Hassan Rejon
Fahad Amjad UTS
By the way we still have 2 days remaining, is what Anthony would say
Let's hope for the best..
8:39 PM

Code Snippet for Business Use Case 02:

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Train an SVM model
svm_model = SVC(random_state=42)
svm_model.fit(X_train_res, y_train_res)

# Make predictions on the validation set
y_pred_svm = svm_model.predict(X_val_res)

# Evaluate the model
accuracy_svm = accuracy_score(y_val_res, y_pred_svm)
precision_svm = precision_score(y_val_res, y_pred_svm)
recall_svm = recall_score(y_val_res, y_pred_svm)
f1_svm = f1_score(y_val_res, y_pred_svm)

print("Validation Set Performance (SVM):")
print(f"Accuracy: {accuracy_svm:.4f}")
print(f"Precision: {precision_svm:.4f}")
print(f"Recall: {recall_svm:.4f}")
print(f"F1 Score: {f1_svm:.4f}")
```

```
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score, roc_curve

# Train an SVM model
svm_model = SVC(random_state=42, probability=True)
svm_model.fit(X_train_res, y_train_res)

# Calculate predictions for the validation set
y_pred_val_svm = svm_model.predict(X_val_res)

# Print confusion matrix and classification report for the validation set
print("Confusion Matrix for Validation Set (SVM):")
print(confusion_matrix(y_val_res, y_pred_val_svm))
print("\nClassification Report for Validation Set (SVM):")
print(classification_report(y_val_res, y_pred_val_svm))
print("\nROC AUC Score for Validation Set (SVM):", roc_auc_score(y_val_res, y_pred_val_svm))

# Plot ROC curve for the validation set
fpr_val_svm, tpr_val_svm, _ = roc_curve(y_val_res, svm_model.predict_proba(X_val_res)[:,1])
plt.figure(figsize=(8, 6))
plt.plot(fpr_val_svm, tpr_val_svm, marker='.')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve for Validation Set (SVM)')
plt.show()
```

```python
from sklearn.model_selection import RandomizedSearchCV
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score
import numpy as np

# Define the hyperparameters grid
param_dist = {
    'C': np.logspace(10),  # Regularization parameter
    'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],  # Kernel type
    'gamma': ['scale', 'auto'],  # Kernel coefficient for 'rbf', 'poly', and 'sigmoid'
}

# Create a randomized search object
random_search_svm = RandomizedSearchCV(SVC(random_state=42, probability=True), param_distributions=

# Perform randomized search on the training data
random_search_svm.fit(X_train_res, y_train_res)

# Get the best hyperparameters
best_params_svm = random_search_svm.best_params_
print("Best Hyperparameters for SVM:", best_params_svm)

# Get the best model
best_model_svm = random_search_svm.best_estimator_

# Make predictions on the validation set using the best model
y_pred_val_best_svm = best_model_svm.predict(X_val_res)
```

```python
from sklearn.tree import DecisionTreeClassifier

# Train a Decision Tree model
tree_model = DecisionTreeClassifier(random_state=42)
tree_model.fit(X_train_res, y_train_res)

# Make predictions on the validation set
y_pred_tree = tree_model.predict(X_val_res)

# Evaluate the model
accuracy_tree = accuracy_score(y_val_res, y_pred_tree)
precision_tree = precision_score(y_val_res, y_pred_tree)
recall_tree = recall_score(y_val_res, y_pred_tree)
f1_tree = f1_score(y_val_res, y_pred_tree)

print("Validation Set Performance (Decision Tree):")
print(f"Accuracy: {accuracy_tree:.4f}")
print(f"Precision: {precision_tree:.4f}")
print(f"Recall: {recall_tree:.4f}")
print(f"F1 Score: {f1_tree:.4f}")

```

```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score, roc_curve

# Train a Decision Tree model
tree_model = DecisionTreeClassifier(random_state=42)
tree_model.fit(X_train_res, y_train_res)

# Calculate predictions for the validation set
y_pred_val_tree = tree_model.predict(X_val_res)

# Print confusion matrix and classification report for the validation set
print("Confusion Matrix for Validation Set:")
print(confusion_matrix(y_val_res, y_pred_val_tree))
print("\nClassification Report for Validation Set:")
print(classification_report(y_val_res, y_pred_val_tree))
print("\nROC AUC Score for Validation Set:", roc_auc_score(y_val_res, y_pred_val_tree))

# Plot ROC curve for the validation set
fpr_val_tree, tpr_val_tree, _ = roc_curve(y_val_res, tree_model.predict_proba(X_val_res)[:,1])
plt.figure(figsize=(8, 6))
plt.plot(fpr_val_tree, tpr_val_tree, marker='.')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve for Validation Set (Decision Tree)')
plt.show()
```

```python
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier

# Define the hyperparameters grid
param_grid = {
    'max_depth': [None, 10, 20, 30, 40, 50],  # Maximum depth of the tree
    'min_samples_split': [2, 5, 10],  # Minimum number of samples required to split a node
    'min_samples_leaf': [1, 2, 4]  # Minimum number of samples required at each leaf node
}

# Create a grid search object
grid_search = GridSearchCV(DecisionTreeClassifier(random_state=42), param_grid, cv=5, scoring='f1')

# Perform grid search on the training data
grid_search.fit(X_train_res, y_train_res)

# Get the best hyperparameters
best_params_tree = grid_search.best_params_
print("Best Hyperparameters:", best_params_tree)

# Get the best model
best_model_tree = grid_search.best_estimator_

# Make predictions on the validation set using the best model
y_pred_val_best_tree = best_model_tree.predict(X_val_res)

# Print confusion matrix and classification report for the validation set using the best model
print("\nConfusion Matrix for Validation Set (Best Model - Decision Tree):")
print(confusion_matrix(y_val_res, y_pred_val_best_tree))
print("\nClassification Report for Validation Set (Best Model - Decision Tree):")
print(classification_report(y_val_res, y_pred_val_best_tree))
print("\nROC AUC Score for Validation Set (Best Model - Decision Tree):", roc_auc_score(y_val_res, y_pred_val_best_tree))
```



```python
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Train a KNN model
knn_model = KNeighborsClassifier()
knn_model.fit(X_train_res, y_train_res)

# Make predictions on the validation set
y_pred_knn = knn_model.predict(X_val_res)

# Evaluate the model
accuracy_knn = accuracy_score(y_val_res, y_pred_knn)
precision_knn = precision_score(y_val_res, y_pred_knn)
recall_knn = recall_score(y_val_res, y_pred_knn)
f1_knn = f1_score(y_val_res, y_pred_knn)

print("Validation Set Performance (KNN):")
print(f"Accuracy: {accuracy_knn:.4f}")
print(f"Precision: {precision_knn:.4f}")
print(f"Recall: {recall_knn:.4f}")
print(f"F1 Score: {f1_knn:.4f}")
```

```python
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score, roc_curve
import matplotlib.pyplot as plt

# Train a KNN model
knn_model = KNeighborsClassifier()
knn_model.fit(X_train_res, y_train_res)

# Calculate predictions for the validation set
y_pred_val_knn = knn_model.predict(X_val_res)

# Print confusion matrix and classification report for the validation set
print("Confusion Matrix for Validation Set:")
print(confusion_matrix(y_val_res, y_pred_val_knn))
print("\nClassification Report for Validation Set:")
print(classification_report(y_val_res, y_pred_val_knn))
print("\nROC AUC Score for Validation Set:", roc_auc_score(y_val_res, y_pred_val_knn))

# Plot ROC curve for the validation set
y_proba_knn = knn_model.predict_proba(X_val_res)[:, 1]
fpr_val_knn, tpr_val_knn, _ = roc_curve(y_val_res, y_proba_knn)
plt.figure(figsize=(8, 6))
plt.plot(fpr_val_knn, tpr_val_knn, marker='.')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve for Validation Set (KNN)')
plt.show()
```

```python
# Define the hyperparameters grid for KNN
param_grid = {
    'n_neighbors': [3, 5, 7, 9, 11],  # Number of neighbors to use
    'weights': ['uniform', 'distance'],  # Weight function used in prediction
    'metric': ['euclidean', 'manhattan', 'minkowski']  # Distance metric
}

# Create a grid search object
grid_search = GridSearchCV(KNeighborsClassifier(), param_grid, cv=5, scoring='f1')

# Perform grid search on the training data
grid_search.fit(X_train_res, y_train_res)

# Get the best hyperparameters
best_params_knn = grid_search.best_params_
print("Best Hyperparameters:", best_params_knn)

# Get the best model
best_model_knn = grid_search.best_estimator_

# Make predictions on the validation set using the best model
y_pred_val_best_knn = best_model_knn.predict(X_val_res)

# Print confusion matrix and classification report for the validation set using the best model
print("\nConfusion Matrix for Validation Set (Best Model - KNN):")
print(confusion_matrix(y_val_res, y_pred_val_best_knn))
print("\nClassification Report for Validation Set (Best Model - KNN):")
print(classification_report(y_val_res, y_pred_val_best_knn))
print("\nROC AUC Score for Validation Set (Best Model - KNN):", roc_auc_score(y_val_res, y_pred_val_best_knn))

# Plot ROC curve for the validation set
y_proba_knn = best_model_knn.predict_proba(X_val_res)[:, 1]
fpr_val_knn, tpr_val_knn, _ = roc_curve(y_val_res, y_proba_knn)
plt.figure(figsize=(8, 6))
```

```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Build the neural network model
model = Sequential()
model.add(Dense(64, input_dim=X_train_res.shape[1], activation='relu'))  # Input layer
model.add(Dense(32, activation='relu'))  # Hidden layer
model.add(Dense(1, activation='sigmoid'))  # Output layer

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_train_res, y_train_res, epochs=10, batch_size=32, validation_data=(X_val_res, y_val_res), verbose=1)

# Make predictions on the validation set
y_pred_nn = model.predict(X_val_res)
y_pred_nn = (y_pred_nn > 0.5).astype(int)

# Evaluate the model
accuracy_nn = accuracy_score(y_val_res, y_pred_nn)
precision_nn = precision_score(y_val_res, y_pred_nn)
recall_nn = recall_score(y_val_res, y_pred_nn)
f1_nn = f1_score(y_val_res, y_pred_nn)

print("Validation Set Performance (Neural Network):")
print(f"Accuracy: {accuracy_nn:.4f}")
print(f"Precision: {precision_nn:.4f}")
print(f"Recall: {recall_nn:.4f}")
print(f"F1 Score: {f1_nn:.4f}")
```

```python
# from sklearn.model_selection import GridSearchCV
from sklearn.neural_network import MLPClassifier

# Define the hyperparameters grid
param_grid_nn = {
    'hidden_layer_sizes': [(100,), (50, 50), (50, 100, 50)],  # Size of hidden layers
    'activation': ['logistic', 'tanh', 'relu'],  # Activation function for the hidden layer
    'solver': ['adam', 'sgd'],  # Solver for weight optimization
    'alpha': [0.0001, 0.001, 0.01]  # L2 penalty (regularization term) parameter
}

# Create a grid search object
grid_search_nn = GridSearchCV(MLPClassifier(random_state=42), param_grid_nn, cv=5, scoring='f1')

# Perform grid search on the training data
grid_search_nn.fit(X_train_res, y_train_res)

# Get the best hyperparameters
best_params_nn = grid_search_nn.best_params_
print("Best Hyperparameters:", best_params_nn)

# Get the best model
best_model_nn = grid_search_nn.best_estimator_

# Make predictions on the validation set using the best model
y_pred_val_best_nn = best_model_nn.predict(X_val_res)

# Print confusion matrix and classification report for the validation set using the best model
print("\nConfusion Matrix for Validation Set (Best Model - Neural Network):")
print(confusion_matrix(y_val_res, y_pred_val_best_nn))
print("\nClassification Report for Validation Set (Best Model - Neural Network):")
print(classification_report(y_val_res, y_pred_val_best_nn))
print("\nROC AUC Score for Validation Set (Best Model - Neural Network):", roc_auc_score(y_val_res, y_pred_val_best_nn))
```

```python
# from sklearn.neural_network import MLPClassifier
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score, roc_curve
import matplotlib.pyplot as plt
from sklearn.neural_network import MLPClassifier
# Train a Neural Network model
nn_model = MLPClassifier(random_state=42)
nn_model.fit(X_train_res, y_train_res)

# Calculate predictions for the validation set
y_pred_val_nn = nn_model.predict(X_val_res)

# Print confusion matrix and classification report for the validation set
print("Confusion Matrix for Validation Set:")
print(confusion_matrix(y_val_res, y_pred_val_nn))
print("\nClassification Report for Validation Set:")
print(classification_report(y_val_res, y_pred_val_nn))
print("\nROC AUC Score for Validation Set:", roc_auc_score(y_val_res, y_pred_val_nn))

# Plot ROC curve for the validation set
fpr_val_nn, tpr_val_nn, _ = roc_curve(y_val_res, nn_model.predict_proba(X_val_res)[:,1])
plt.figure(figsize=(8, 6))
plt.plot(fpr_val_nn, tpr_val_nn, marker='.')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve for Validation Set (Neural Network)')
plt.show()
```