Report name:
Customer Data Management And Analysis

Reported by:

Menna Allah Sayed Ali	1112149599
Basant Waleed Yehia	1112144832
Nadine Ibrahim	1112131247
Yazied Mohamed Hassan	1112131350
Ahmed Mohamed Yousef	1112140141

The project idea and purpose:

This project aims to build a comprehensive system for managing and analyzing customer data, using a variety of tools and technologies. The project is based on a comprehensive approach starting from designing the SQL database all the way to deploying a machine learning model in the production environment.

☐ How was the project implemented over 4 weeks?

Week 1 : Data Management And SQL Database Setup …

Introduction

The first week of the project aims to build a customer data management system by designing and implementing an SQL database. During this week, the database schema will be designed, representing the way data is organized within the database. Additionally, the database will be created and populated with relevant customer data. Finally, SQL queries will be written to extract and analyze data with the goal of better understanding our customers and making informed decisions.

1. Database Design

To design a logical and efficient structure for storing customer data in an organized and easily retrievable manner.

Steps followed:

- Identifying Entities: Key entities to be represented in the database were identified, such as:
- Customers table: Contains basic information about each customer (name, email, address, phone number, etc.).
- Transactions table: Records all transactions made by customers (transaction date, products purchased, total value, etc.).
- Interactions table: Records all interactions between customers and the company (emails, calls, surveys, etc.).

* Defining Relationships:Relationships between different entities were defined, such as the one-to-many relationship between the customers table and the transactions table (one customer can make many transactions).

* Defining Fields:The necessary fields for each table were defined, ensuring the selection of appropriate data types (text, number, date, etc.).

* Defining Keys:Primary and composite keys were defined for each table to ensure data integrity and uniqueness of records.

2. Database Implementation

To create the actual database and populate it with data.

-Tools used: Microsoft SQL Server and SQL Management Studio.

Steps followed:

I. Creating the database:A new database was created on the SQL Server.
II. Creating tables:Tables were created according to the designed schema.

III. Defining relationships:Relationships between tables were defined using foreign keys.
IV. Populating data:Initial data was entered into the tables using the management tools available in SQL Management Studio.

3. Writing SQL Queries

To extract and analyze data stored in the database.

Types of queries:

a. Retrieval queries:To extract specific data from tables (e.g., extracting a list of customer names who purchased a specific product).
b. Update queries:To modify existing data in tables (e.g., updating the address of a specific customer).
c. Delete queries:To delete unnecessary data from tables.
d. Analysis queries:To analyze data and produce reports (e.g., calculating total sales for each product).

Accomplishments

i) A comprehensive and efficient database schema was designed for customer data management.
ii) An SQL database was created on Microsoft SQL Server and populated with initial data.
iii) A set of SQL queries was written to extract and analyze data.

Summary

- Data Sourcing: Utilized datasets sourced from Kaggle to obtain real-world customer data.
- Database Design and schema : Designed a robust schema to manage customer data.

Tables Created:

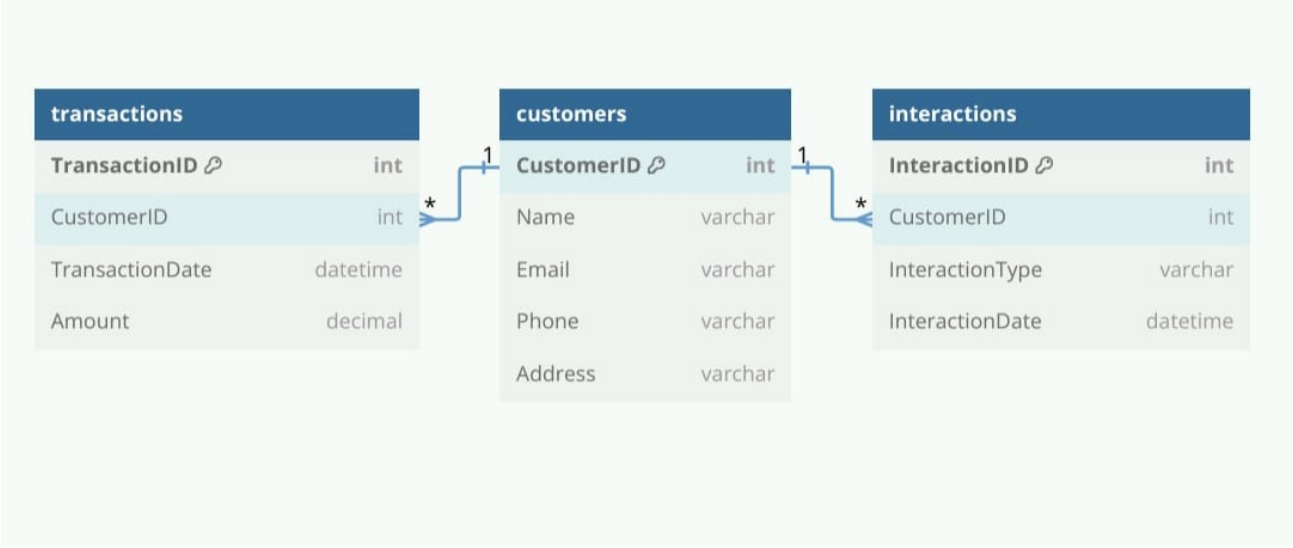
- telco_customer_data: Contains comprehensive customer

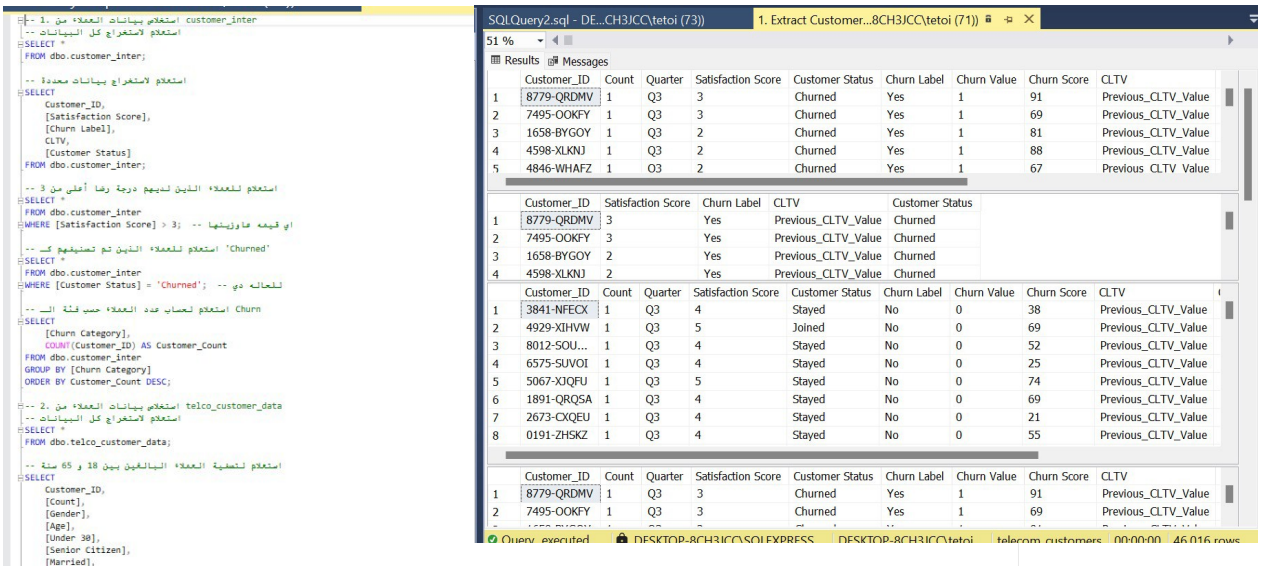
information.

- customer_trans: Tracks transaction details for each customer.
- customer_inter: Captures customer interaction records.

• Implementation: Successfully created and populated the database using Microsoft SQL Server, ensuring data integrity and accessibility.

Delivered a well-structured database schema and developed SQL queries for efficient data extraction and analysis





```sql
-- 1. استعلام عن بيانات العملاء من customer_inter
-- استعلام لاستخراج كل البيانات
SELECT *
FROM dbo.customer_inter;

-- استعلام لاستخراج بيانات محددة
SELECT
    Customer_ID,
    [Satisfaction Score],
    [Churn Label],
    CLTV,
    [Customer Status]
FROM dbo.customer_inter;

-- استعلام للعملاء الذين لديهم درجة رضا أعلى من 3
SELECT *
FROM dbo.customer_inter
WHERE [Satisfaction Score] > 3;  -- اي قيمه فاوزيتها

-- استعلام للعملاء الذين تم تصنيفهم كـ 'Churned'
SELECT *
FROM dbo.customer_inter
WHERE [Customer Status] = 'Churned';  -- للعمالله دي

-- استعلام لحساب عدد العملاء حسب فئة الـ Churn
SELECT
    [Churn Category],
    COUNT(Customer_ID) AS Customer_Count
FROM dbo.customer_inter
GROUP BY [Churn Category]
ORDER BY Customer_Count DESC;

-- 2. استعلام عن بيانات العملاء من telco_customer_data
-- استعلام لاستخراج كل البيانات
SELECT *
FROM dbo.telco_customer_data;

-- استعلام لتصفية العملاء البالغين بين 18 و 65 سنة
SELECT
    Customer_ID,
    [Count],
    [Gender],
    [Age],
    [Under 30],
    [Senior Citizen],
    [Married],
```

SQLQuery2.sql - DE...CH3JCC\tetoi (73)) | 1. Extract Customer...8CH3JCC\tetoi (71))

51 %

Results | Messages

	Customer_ID	Count	Quarter	Satisfaction Score	Customer Status	Churn Label	Churn Value	Churn Score	CLTV
1	8779-QRDMV	1	Q3	3	Churned	Yes	1	91	Previous_CLTV_Value
2	7495-OOKFY	1	Q3	3	Churned	Yes	1	69	Previous_CLTV_Value
3	1658-BYGOY	1	Q3	2	Churned	Yes	1	81	Previous_CLTV_Value
4	4598-XLKNJ	1	Q3	2	Churned	Yes	1	88	Previous_CLTV_Value
5	4846-WHAFZ	1	Q3	2	Churned	Yes	1	67	Previous_CLTV_Value

	Customer_ID	Satisfaction Score	Churn Label	CLTV	Customer Status
1	8779-QRDMV	3	Yes	Previous_CLTV_Value	Churned
2	7495-OOKFY	3	Yes	Previous_CLTV_Value	Churned
3	1658-BYGOY	2	Yes	Previous_CLTV_Value	Churned
4	4598-XLKNJ	2	Yes	Previous_CLTV_Value	Churned

	Customer_ID	Count	Quarter	Satisfaction Score	Customer Status	Churn Label	Churn Value	Churn Score	CLTV
1	3841-NFECX	1	Q3	4	Stayed	No	0	38	Previous_CLTV_Value
2	4929-XIHVW	1	Q3	5	Joined	No	0	69	Previous_CLTV_Value
3	8012-SOU...	1	Q3	4	Stayed	No	0	52	Previous_CLTV_Value
4	6575-SUVOI	1	Q3	4	Stayed	No	0	25	Previous_CLTV_Value
5	5067-XJQFU	1	Q3	5	Stayed	No	0	74	Previous_CLTV_Value
6	1891-QRQSA	1	Q3	4	Stayed	No	0	69	Previous_CLTV_Value
7	2673-CXQEU	1	Q3	4	Stayed	No	0	21	Previous_CLTV_Value
8	0191-ZHSKZ	1	Q3	4	Stayed	No	0	55	Previous_CLTV_Value

	Customer_ID	Count	Quarter	Satisfaction Score	Customer Status	Churn Label	Churn Value	Churn Score	CLTV
1	8779-QRDMV	1	Q3	3	Churned	Yes	1	91	Previous_CLTV_Value
2	7495-OOKFY	1	Q3	3	Churned	Yes	1	69	Previous_CLTV_Value

Query executed... | DESKTOP-8CH3JCC\SQLEXPRESS | DESKTOP-8CH3JCC\tetoi | telecom_customers | 00:00:00 | 46,016 rows

Results | Messages

6	5777-ZPQNC	1	Q3	0	0	12	Off...	1	26.67
7	1951-IEYXM	1	Q3	1	1	72	None	1	34.97
8	3318-NMQXL	1	Q3	0	0	3	Off...	1	15.6
9	1022-RKXDR	1	Q3	0	0	41	None	1	32.66
10	2361-FJWNO	1	Q3	0	0	40	None	0	0
11	2272-UOINI	1	Q3	0	0	7	Off...	1	3.34
12	8232-UTFOZ	1	Q3	0	0	69	None	1	7.18
13	3750-YHRYO	1	Q3	1	1	7	None	1	21.09

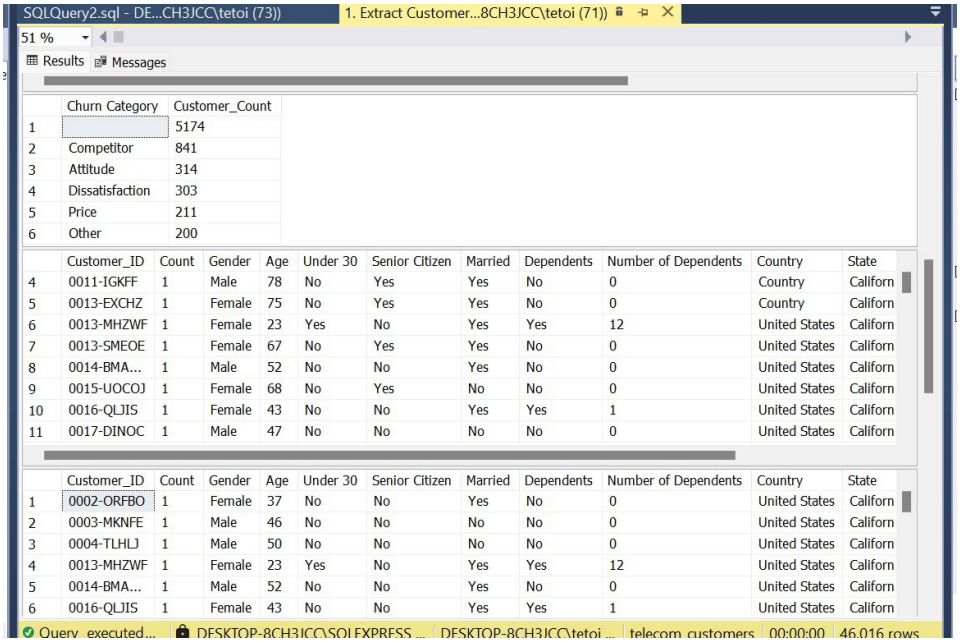
Click to Select the whole row

	Customer_ID	Monthly Charge	Total Revenue	Contract
8	3318-NMQXL	111.1	313.6	Month-to-Month
9	1022-RKXDR	24.85	2301.31	One Year
10	2361-FJWNO	36	1512.9	One Year
11	2272-UOINI	95.7	594.43	Month-to-Month
12	8232-UTFOZ	19.95	1894.77	Two Year
13	3750-YHRYO	20.65	297.63	One Year
14	6637-KYRCV	37.4	167.2	Month-to-Month
15	5668-MEISB	106.1	8118.92	Two Year

	Avg_Monthly_Charge	Total_Revenue
1	71.911543	21558123

	Total_Customers	Total_Referrals
1	3222	13747

	Quarter	Customer_Count
1	Q3	7043



SQLQuery2.sql - DE...CH3JCC\tetoi (73)) | 1. Extract Customer...8CH3JCC\tetoi (71))

51 %

Results | Messages

	Churn Category	Customer_Count
1		5174
2	Competitor	841
3	Attitude	314
4	Dissatisfaction	303
5	Price	211
6	Other	200

	Customer_ID	Count	Gender	Age	Under 30	Senior Citizen	Married	Dependents	Number of Dependents	Country	State
4	0011-IGKFF	1	Male	78	No	Yes	Yes	No	0	Country	Californ
5	0013-EXCHZ	1	Female	75	No	Yes	Yes	No	0	Country	Californ
6	0013-MHZWF	1	Female	23	Yes	No	Yes	Yes	12	United States	Californ
7	0013-SMEOE	1	Female	67	No	Yes	Yes	No	0	United States	Californ
8	0014-BMA...	1	Male	52	No	No	Yes	No	0	United States	Californ
9	0015-UOCOJ	1	Female	68	No	Yes	No	No	0	United States	Californ
10	0016-QLJIS	1	Female	43	No	No	Yes	Yes	1	United States	Californ
11	0017-DINOC	1	Male	47	No	No	No	No	0	United States	Californ

	Customer_ID	Count	Gender	Age	Under 30	Senior Citizen	Married	Dependents	Number of Dependents	Country	State
1	0002-ORFBO	1	Female	37	No	No	Yes	No	0	United States	Californ
2	0003-MKNFE	1	Male	46	No	No	No	No	0	United States	Californ
3	0004-TLHLJ	1	Male	50	No	No	No	No	0	United States	Californ
4	0013-MHZWF	1	Female	23	Yes	No	Yes	Yes	12	United States	Californ
5	0014-BMA...	1	Male	52	No	No	Yes	No	0	United States	Californ
6	0016-QLJIS	1	Female	43	No	No	Yes	Yes	1	United States	Californ

Query executed... | DESKTOP-8CH3JCC\SQLEXPRESS ... | DESKTOP-8CH3JCC\tetoi ... | telecom_customers | 00:00:00 | 46,016 rows

Week 2 : Data Warehousing and Python Programming

Introduction

This week's focus was on establishing an SQL data warehouse and developing Python programs to interact with this warehouse and prepare data for analysis.

* SQL Data Warehouse: A Microsoft SQL Data Warehouse was implemented to collect and manage large volumes of customer data for analysis and insight extraction.
* Data Integration: Data from various sources was loaded into the data warehouse. The integration process involved data transformation and cleansing to ensure quality and consistency.
* Python Programming: A suite of Python programs was developed to interact with the database. These programs extracted and prepared data for analysis. Python libraries such as Pandas and SQLAlchemy were used to facilitate these processes.

Tools Used

1. Microsoft SQL Data Warehouse: A robust tool for creating and managing data warehouses.
2. Python: A versatile programming language widely used in data analysis.
3. Pandas: A Python library for data analysis and manipulation.
4. SQLAlchemy: A Python library for interacting with various databases, including SQL Server.

Outcomes:

Efficient SQL Data Warehouse: The warehouse contains integrated data ready for analysis.

Python Programs for Data Extraction and Preparation: These programs can be used to automate future analysis processes.

Summary

- Data Warehouse Implementation:

  Built a SQL Data Warehouse to aggregate and manage large volumes of customer data for analytical insights.

- Schema Overview:

  Fact Table: customer_trans

  Primary Key: trans_ID (Surrogate Key)

  Business Key: trans_Customer_ID

- Dimension Tables:

  - dbo.customer_inter

    Primary Key: interaction_ID

  - dbo.telco_customer_data

    Primary Key: Customer_Telco_ID

- Importance: This structure allows for efficient querying and

analysis, enabling better insights into customer behavior.

Schema Overview

• Fact Table: customer_trans

Primary Key: trans_ID (Surrogate Key)

Business Key: trans_Customer_ID

Foreign Keys:

inter_Customer_ID → References dbo.customer_inter(inter_Customer_ID)

Customer_Telco_ID → References dbo.telco_customer_data(Customer_Telco_ID)

• Dimension Table 1: dbo.customer_inter

Primary Key: interaction_ID (Surrogate Key)

Business Key: inter_Customer_ID

• Dimension Table 2: dbo.telco_customer_data

Primary Key: Customer_Telco_ID (Surrogate Key)

Business Key: Customer_ID

Keys Overview

Surrogate Keys:

trans_ID (Primary Key for customer_trans)

interaction_ID (Primary Key for dbo.customer_inter)

Customer_Telco_ID (Primary Key for dbo.telco_customer_data)

• Business Keys:

trans_Customer_ID (Business Key in customer_trans)

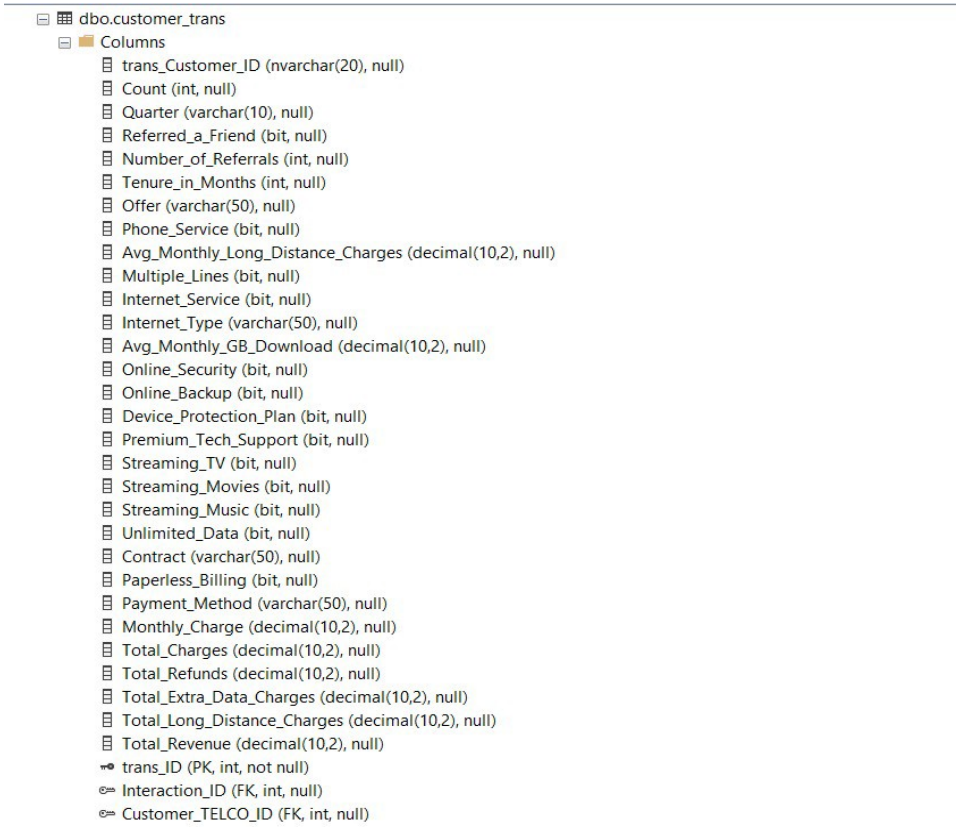
inter_Customer_ID (Business Key in dbo.customer_inter)

Customer_ID (Business Key in dbo.telco_customer_data)

Deliverables

A fully functional SQL Data Warehouse along with Python scripts for data extraction and preparation.


Statistics
dbo.telco_customer_data
Columns
Customer_ID (nvarchar(20), not null)
Count (int, null)
Gender (varchar(10), null)
Age (int, null)
Under_30 (bit, null)
Senior_Citizen (bit, null)
Married (bit, null)
Dependents (bit, null)
Number_of_Dependents (int, null)
Country (varchar(50), null)
State (varchar(50), null)
City (varchar(50), null)
Zip_Code (varchar(10), null)
Customer_TELCO_ID (PK, int, not null)
Keys
Customer_TELCO_ID


dbo.customer_trans
Columns
trans_Customer_ID (nvarchar(20), null)
Count (int, null)
Quarter (varchar(10), null)
Referred_a_Friend (bit, null)
Number_of_Referrals (int, null)
Tenure_in_Months (int, null)
Offer (varchar(50), null)
Phone_Service (bit, null)
Avg_Monthly_Long_Distance_Charges (decimal(10,2), null)
Multiple_Lines (bit, null)
Internet_Service (bit, null)
Internet_Type (varchar(50), null)
Avg_Monthly_GB_Download (decimal(10,2), null)
Online_Security (bit, null)
Online_Backup (bit, null)
Device_Protection_Plan (bit, null)
Premium_Tech_Support (bit, null)
Streaming_TV (bit, null)
Streaming_Movies (bit, null)
Streaming_Music (bit, null)
Unlimited_Data (bit, null)
Contract (varchar(50), null)
Paperless_Billing (bit, null)
Payment_Method (varchar(50), null)
Monthly_Charge (decimal(10,2), null)
Total_Charges (decimal(10,2), null)
Total_Refunds (decimal(10,2), null)
Total_Extra_Data_Charges (decimal(10,2), null)
Total_Long_Distance_Charges (decimal(10,2), null)
Total_Revenue (decimal(10,2), null)
trans_ID (PK, int, not null)
Interaction_ID (FK, int, null)
Customer_TELCO_ID (FK, int, null)

Week 3: Data Science and Azure Integration

Introduction

We focuses in this section on data science and its integration with Azure services. Through the implementation of a series of tasks related to data analysis, the development of predictive models using Python, and leveraging Azure Data services for data management, analysis, and the development and evaluation of machine learning models using Azure Machine Learning, significant progress has been made. The outputs of this week include a comprehensive analytical report containing extracted insights and predictive models, as well as the setup and documentation of an integrated Azure Data services environment.

1. Data Science with Python

* Data Analysis: In-depth analysis of the data was conducted to understand its nature, structure, and underlying relationships and patterns. A variety of statistical and visual techniques were employed to achieve this.
* Predictive Model Development: Predictive models were built using various machine learning algorithms such as logistic regression, decision trees, and random forests. The most suitable algorithm was selected based on the nature of the data and the problem being solved.

Model Evaluation: Model performance was evaluated using a variety of metrics such as accuracy, sensitivity, specificity, and F1-score.

2. Azure Data Fundamentals

* Data Management: Various Azure Data services were used to effectively manage data, including data storage, organization, and retrieval.
* Data Analysis: Complex data analyses were performed using Azure Data tools to gain deeper insights.

3. Model Development

* Model Development: Machine learning models were developed using Azure Machine Learning or similar tools.
* Model Evaluation: Models were evaluated to ensure their accuracy and generalization capabilities.

4. Tools Used

* Python: The primary programming language used for data analysis and model development.
* Scikit-learn: A Python library for machine learning.
* Matplotlib: A Python library for creating visualizations.
* Azure Data Studio: A tool for managing and exploring data in Azure.
* Azure Machine Learning: A platform for developing and deploying machine learning models.

Outputs

* Analytical Report: A comprehensive report was prepared that includes the main results obtained during the analysis process, the extracted insights, and the developed predictive models.
* Integrated Azure Data Environment: An integrated environment was created on Azure for data storage, analysis, and management.

Summary

• Data Science with Python:

we focused on the integration of data science and Azure services.

we explored data science with Python, conducting comprehensive

data analysis and predictive modeling

• Azure Integration:

Leveraged Azure services : Additionally, we leveraged Azure

services for efficient data management and analytics..

• Model Development:

Developed and evaluated machine learning models to predict

customer behavior based on transaction and interaction data.

Data Science with Python

• Churn prediction to identify at-risk customers:

Churn prediction was a key aspect of our data science work. By analyzing transaction and interaction data, we were able to identify customers who were likely to churn. This allowed us to proactively implement retention strategies and minimize customer attrition.

• Conducting comprehensive data analysis and predictive modeling: we conducted in-depth data analysis and built predictive models using Python. One notable application was churn prediction, which helped us identify at-risk customers. These models provided valuable insights into customer behavior and enabled targeted retention strategies.

Deliverables

Generated a detailed analysis report outlining insights and presented predictive models to stakeholders.

1) Read the Three CSV Tables

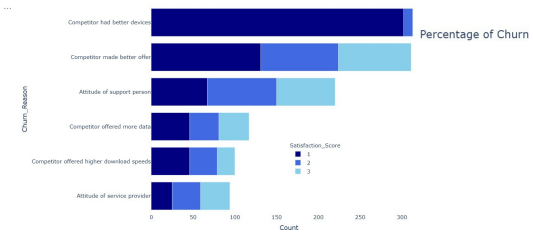
Read the paths of our three csv tables

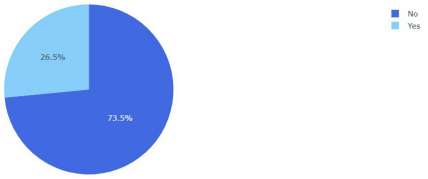
+ Code + Markdown

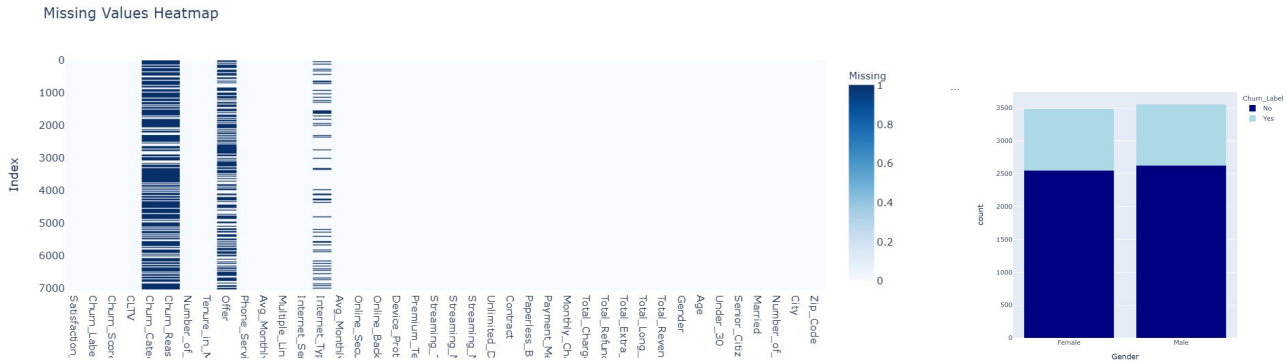
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
%matplotlib inline
warnings.filterwarnings("ignore")
from azureml.core import Dataset, Workspace

# Connect to your Azure ML workspace
ws = Workspace.from_config()

# Get the dataset (the folder you've uploaded in Data Assets)
dataset = Dataset.File.from_files(path=(ws.datastores['workspaceblobstore'], 'UI/2024-10-17_090040_UTC/dw data/*'))

# Display the file paths in the dataset
file_paths = dataset.to_path()
print(file_paths)
```

[64] ✓ <1 sec

```
{'infer_column_types': 'False', 'activity': 'to_path'}
```







	Missing_Number	Missing_Percent
Offer	3877	0.550476
Internet_Type	1526	0.216669
Satisfaction_Score	0	0.000000
Total_Long_Distance_Charges	0	0.000000
Contract	0	0.000000
Paperless_Billing	0	0.000000
Payment_Method	0	0.000000
Monthly_Charge	0	0.000000
Total_Charges	0	0.000000
Total_Refunds	0	0.000000

```
from IPython.display import display
# Iterate over the file paths to read each CSV file
dfs = []

# Make the static path or the folder path that will be the same in the three files
static_path = "azureml://subscriptions/0146ae78-9468-4285-8cab-97231deb201d/resourcegroups/Final_Project/workspaces/CustomerChurnPredict

for file_path in file_paths:
    if file_path.endswith('.csv'):
        # Read the CSV file directly from the blob storage into a pandas DataFrame
        df = pd.read_csv(static_path.format(file_path))
        dfs.append(df)

# Check the number of dataframes loaded
print(f"Successfully Loaded {len(dfs)} CSV files.")

# Display the first few rows of each DataFrame to verify
print("\ncustomer_inter.csv:")
display(dfs[0].head())
print(dfs[0].shape)

print("\ncustomer_trans.csv:")
display(dfs[1].head())
print(dfs[1].shape)

print("\ntelco_customer_data.csv:")
display(dfs[2].head())
print(dfs[2].shape)
```

Week 4:MLOps , Deployment.

Introduction

Week four marked a pivotal transition from model development to deployment. This week, we focused on MLOps aspects and model deployment.

* MLOps: Experiment Tracking and Model Management.
* MLflow: The MLflow platform was employed to track various experiments conducted on the model. This facilitated a deeper understanding of performance and streamlined the process of selecting the optimal model.
* Model Management: MLflow was also utilized to manage the diverse trained models, enhancing collaboration among team members and enabling easy access to previous models.

Model Deployment

* Azure Services: Azure services were selected as the deployment platform due to their flexibility and robustness. The model was successfully deployed on one of these services to deliver predictions in a production environment.

Tools Used

* MLflow: For experiment tracking and model management.
* Azure Services: For model deployment.

Achievements

* Deployed Machine Learning Model: The machine learning model was successfully deployed on Azure services…

Summary

- MLOps Implementation:

  Integrating MLflow for tracking experiments and version control

  We implemented MLOps practices by integrating MLflow into our workflow. MLflow allowed us to track machine learning experiments, manage model versions, and efficiently collaborate with team members. This ensured reproducibility and traceability throughout the project.

- Deployment:

  Deploying ML models using Azure or web applications (Flask or Streamlit)We deployed our machine learning models using Azure

services or web applications such as Flask or Streamlit. This allowed us to provide user-friendly access to our predictive insights. By deploying our models, we made our solutions accessible and actionable for stakeholders.

Deliverables Our final deliverables included the deployment of our machine learning models or web applications. Alongside that, we provided a comprehensive final project report summarizing our methodology, findings, and recommendations.

Project Objectives

- Data Management: Design a well-structured database that can efficiently store, retrieve, and update customer data.
- Predictive Analysis: Develop machine learning models to predict customer churn, helping to identify customers at risk of leaving.
- Data Warehousing: Implement a data warehouse to centralize customer data, supporting analysis and reporting.
- Deployment: Deploy a predictive model accessible through a web application or Azure, providing actionable insights.

Key Insights

- Customer Management: Improved data management and analysis through SQL and Python, reducing manual processes and errors.
- Churn Prediction: Machine learning models provided insights into at-risk customers, allowing targeted retention

strategies.

- Scalability: Azure integration and data warehousing created a scalable solution for handling large datasets and real-time analytics.

Project Benefits

- Enhanced Decision-Making: The predictive model supports strategic decision-making by identifying potential churn risks.
- Efficient Data Management: SQL and Azure-based solutions streamlined data handling, reducing time spent on manual tasks.
- Scalability and Future Growth: The data warehouse and Azure setup enable future scaling, with the infrastructure supporting larger datasets and new models.

Conclusion

This project successfully achieved a comprehensive solution for customer data management, data analysis, and predictive modeling. The deployed machine learning model provides actionable insights, enabling the telecom company to retain customers and optimize decision-making processes