CUSTOMER DATA MANAGEMENT AND ANALYSIS

DEPI GRADUATION PROJECT
MICROSOFT DATA ENGINEER

Supervior
Eng. Amira Youssef

Group code
CAI1_AIS4_S10d

OCTOBER 2024

00

PARTICIPANTS

1. Susana Ayman
2. Shahd Mustafa
3. Nour Hesham
4. Muhammad Yasser
5. Khaled Omar
6. Ahmed Mhmoud Adly

00

TABLE OF CONTENTS

01

INTRODUCTION

Project's purpose:

The ability to analyze customer behavior and preferences are critical for companies looking to remain competitive. By effectively leveraging customer data, businesses can unlock valuable insights that drive personalized marketing strategies, enhance customer retention, and optimize overall business performance, empowers organizations to make informed decisions, anticipate customer needs, and foster long-term relationships.

Goals and objectives:

This project focuses on developing a comprehensive system for customer data management and analysis. The primary objectives of the project are

1. to efficiently organize and collect customer data from various sources and files to create a data warehouse
2. discover useful insights through data analysis.
3. and deploy predictive models to enhance customer targeting and engagement.

To achieve these objectives, we implemented a range of modern technologies including :

relational databases, SQL queries for data manipulation, Python for advanced data processing, and Azure for cloud integration and machine learning deployment.

02

DATABASE SETUP AND SQL QUERIES

TOOLS USED:

To efficiently manage and analyze customer data, we utilized :

> SQL Server Management Studio (SSMS)

> SQL Server (SQLS)

as the primary tools for database setup and query execution.

>unfiltered data available on Kaggle, to source the data from and populate the database

Database Schema:

Our schema was designed to encompass key business entities, including:

- **Customers:** Included fields such as customer ID, name, gender, age.
- **Products:** Stored product information, including product ID, name, category, and price.
- **Purchases:** Captured transaction details, such as purchase ID, customer ID, product ID, quantity, and purchase date.
- **Payment Methods:** Contained details about various payment types associated with each purchase.
- **Customer Churn:** Tracked whether a customer was still active or had stopped purchasing (binary churn indicator).

.

SQL Queries:

Once the schema was populated with the Kaggle data, we implemented several SQL queries to extract actionable insights:

1. **Customer Data Extraction**: We wrote queries to extract detailed information about each customer by joining relevant tables (customers, purchases, and payment methods). This allowed us to analyze customer purchase history, and payment preferences.
2. **Product Price Updates:** To reflect real-time changes in product prices, we implemented an **update** query that modified product prices based on certain conditions.
3. **Customer Segmentation by Gender:** We analyzed the customer base by segmenting them according to gender, providing insights into how gender might influence purchasing behavior.

Results:

The SQL queries yielded several important insights:

- **Customer Profile Insights:** Detailed extraction of customer data help in better understanding customer preferences, common product purchases.
- **Product Pricing Management:** The price update queries ensure the product prices reflected real-time market conditions.
- **Purchase Behavior Analysis:** Averaging the purchase values for each customer allow us to identify high-value customers and tailor loyalty programs accordingly.

03

DATA WAREHOUSING & PYTHON PROGRAMMING

Tools Used

- **Microsoft SQL Data Warehouse:** For storing and managing large volumes of customer data.
- **SQL Server Integration Services (SSIS):** For automating data integration and ETL processes.
- **Python (Pandas, SQLAlchemy):** For data extraction, transformation, and advanced data manipulation.
- **Matplotlib:** For creating data visualizations that helped communicate insights effectively.

Data Warehouse Implementation

We set up a Microsoft SQL Data Warehouse to serve as a centralized repository for aggregating customer data from multiple sources. The data warehouse was designed to support high-volume data operations, enabling efficient storage and retrieval for analytics.

The schema in the data warehouse mirrored the structure of the original database (customers, products, purchases, payment methods, and churn data) but was optimized for analytical queries. This allowed us to run complex reports and analyses without affecting transactional systems.

Data Integration

To load and manage data in the warehouse, we used **SQL Server Integration Services (SSIS)**. SSIS facilitated the **ETL** (Extract, Transform, Load) process, allowing us to extract data from various sources, transform it as necessary (e.g., cleaning, filtering), and load it into the warehouse. This process ensured data consistency and integrity across all stages of analysis.
We pulled data from:

- The SQL database containing raw transactional data
- External CSV files sourced from Kaggle
- Additional datasets relevant to customer churn and behavior from flat and excel files

Python Programming

After populating the data warehouse, Python was used to perform further data preparation, and analytical tasks. We primarily used the **Pandas** library for data manipulation and **SQLAlchemy** to connect Python scripts with the SQL data warehouse and **Matpliotlib** for data visualization.
Python tasks included:

1. **Data Extraction:** Using **SQLAlchemy**, we developed scripts to extract relevant data from the warehouse for analysis.
2. **Data Preparation:** We used Pandas to clean duplicate data and null values and prepare the data for analysis.
3. **Data Visualization:** To present the results, we used Matplotlib for visualizing key trends and insights. This included plotting customer churn rates, purchase behavior of the customer.

Results

The integration of a SQL data warehouse and Python programming provided several benefits:

- **Centralized Data Management:** The data warehouse streamlined access to customer data, allowing us to handle large datasets efficiently and run complex analyses .
- **Automated Data Loading:** SSIS automated the process of updating the data warehouse with new records.
- **Data Preparation for Analysis:** Python scripts enabled quick data extraction and preparation.
- **Actionable Insights**: Using Matplotlib, we visualized customer behavior patterns, helping to better understand customer churn.

04

DATA SCIENCE AND AZURE INTEGRATION

In this phase of the project, we focused on building predictive models to analyze customer churn and deployed these models using **Azure cloud services**. By using **machine learning algorithms** and **Azure's cloud infrastructure,** we were able to generate valuable insights into customer behavior and make accurate predictions regarding customer churn.

Data Science:
We used two key machine learning models for churn prediction:

- **Logistic Regression Model:** This was used to explore the relationship between various customer attributes
- **Random Forest Model:** used to capture complex patterns in the data and improve prediction accuracy. The model was trained using customer demographics, purchase behavior, and payment methods to predict the likelihood of customer churn.

Both models were implemented using **Scikit-learn (sklearn) in Python**. After preprocessing the data with **Pandas**, we trained and tested the models on historical customer data.

Azure Integration

To manage data and deploy models, we integrated Azure cloud services into our workflow:

- **Azure Storage Blob:** We used Azure Blob Storage in Python to store and retrieve large datasets.
-
- **Model Deployment on Azure:** The trained machine learning models were deployed using Azure Machine Learning services. This allowed us to serve the models via a REST API, making them accessible for real-time predictions in production environments.

Results

- **Churn Prediction:** The Random Forest model outperformed the regression model in predicting customer churn **achieving an accuracy of over 80%**. This model was deployed for real-time churn predictions.

- **Azure Integration:** Azure's storage and machine learning services allowed seamless deployment, providing a flexible and scalable environment for storing data and deploying models for real-time use.

05

MLOPS, DEPLOYMENT, AND FINAL PRESENTATION.

In the final phase of the project, we focused on deploying our machine learning models and creating a user-friendly interface for real-time churn prediction. This was achieved through **the implementation of MLOps** and **the development of a web application** to ensure the models were accessible and functional in a live environment.

Model Management with MLflow

We integrated MLflow to track, register, and manage The random forest model (which demonstrated the highest accuracy in churn prediction) and the regression model were uploaded and registered in MLflow, allowing us to:

- **Version the models:** Ensuring consistent management of updates and improvements over time.
- **Track model performance:** Recording metrics such as accuracy, precision, and recall, to monitor the model's predictive performance.
- **Register models for deployment:** Making the models easily accessible for production deployment via streamlined registration in MLflow.

Web Application Deployment with Streamlit
To make the churn prediction models accessible to end-users, we developed a web application using Streamlit.

Key features of the application included:

- **User Input Interface:** Users can input key customer details (e.g., purchase history, product interactions, demographics) via a simple interface.
- **Real-time Predictions:** Once the data is submitted, the model runs in the backend to predict the likelihood of customer churn.
- **Visualization of Results and analysis:** The app also provides visual feedback using Matplotlib, showing key useful insights such as churn analysis , products and pricicng analysis, seasonal trends.

Results

- **Operationalized Churn Prediction**: The deployment of the random forest model in a web-based interface allowed for real-time customer churn prediction, achieving an accuracy of over 80%. Business users could now predict customer churn quickly.
- **Seamless Model Management:** With MLflow, we ensured efficient tracking, versioning, and deployment of our models, simplifying the management of the model.
- **Enhanced User Experience:** The Streamlit-based web application provided a simple yet powerful tool for business users to make data-driven decisions based on the model's outputs.

09

CONCLUSIONS

This project in customer data management and analysis successfully integrated modern tools and methodologies to **manage, analyze, and extrac**t insights from large datasets.

- By implementing a **SQL Data Warehouse,** we centralized customer data, ensuring efficient management and scalability for analytical purposes.
- **The use of Python** for data extraction, preparation, and visualization streamlined our analysis, while SQL queries provided immediate insights into customer behaviors and trends.
- **Using data science** we employed regression models to predict customer churn, demonstrating high accuracy.
- The integration with **Azure services** allowed us to deploy these models in a scalable and reliable environment, ensuring real-time predictions for business decision-making.

The project demonstrated the value of data warehousing, machine learning, and cloud integration in improving customer insights and predictive capabilities, laying a foundation for more sophisticated data-driven strategies in the future.