وزارة الاتصـــــــالات
وتكنولوجيا المعلومات

رواد مصر الرقمية

Customer Data Management and Analysis

Introduction:

Overview

The "Customer Data Management and Analysis" project focuses on building an end-to-end solution for managing and analyzing customer data. As customer data plays a pivotal role in driving business insights and decisions, this project aims to develop a robust infrastructure that enables efficient data storage, processing, and analysis, leading to actionable insights.

Week 1: Data Management and SQL Database Setup

We started the searching for a dataset that is suitable for the project, then we started in designing the schema for the tables of the dataset, The tables show the customers data, Transactions and interactions.

We Used the Bulk insert function to implement the data into the database.

Week 2: Data Warehousing and Python Programming

We analyzed customer data within a database using SQL queries and created a data warehouse schema to improve data organization and analysis. This included designing fact and dimension tables tailored to specific business needs. For data integration, we wrote SQL queries to load data from operational databases into the warehouse tables, maintaining data consistency and integrity. Additionally, I worked with CSV files, transforming and loading the data into the warehouse using SSIS, managing duplicates, and performing necessary transformations to fit the data warehouse structure.

Python script: We wrote a python script to interact with SQL Server and extract data into csv.

Week 3: Data Science and Azure Integration Data science:

Data Exploration and Preprocessing:

Imported necessary libraries and loaded train/test datasets

Performed initial data exploration (shape, columns, summary statistics)

Handled duplicates and checked for missing values

Visualized target variable distribution and feature distributions

Data Visualization:

Created various plots to understand feature distributions and relationships

Analyzed correlations between features

Custom Transformers:

Implemented custom transformer classes for scaling, encoding, and

imputation

Data Pipeline:

Created a full pipeline combining various preprocessing steps

Applied the pipeline to transform training and test data

Feature Importance:

Used Random Forest to determine feature importance

Visualized feature importance

Machine Learning Modeling:

Trained and evaluated multiple models ✓Decision Tree, Random Forest, Gradient Boosting, etc.)

Compared model performance using accuracy, AUC, and F1-score

Visualized model performance comparisons

Final Model Selection:

Chose CatBoostClassifier as the best performing model

Plotted confusion matrix and ROC curve for the selected model

Cross-Validation:

Performed 10-fold cross-validation to assess model stability

Hyperparameter Tuning:

Used RandomizedSearchCV to find optimal hyperparameters for CatBoostClassifier

Submission File Creation:

Trained the final model on the entire training set

Made predictions on the test set

Created and saved a submission file

The notebook follows a comprehensive machine learning workflow, from data exploration and preprocessing to model selection, evaluation, and final prediction.

The CatBoostClassifier was chosen as the best model based on AUC score, and

hyperparameter tuning was performed to optimize its performance.

Azure data fundamentals: developing an Automated Machine Learning (AutoML) solution for predicting bank churn, aiming to identify customers likely to exit the bank. We began by setting up the data flow in Azure Data Factory, where we transformed the source dataset by dropping irrelevant columns to enhance model performance. The cleaned data was then directed to a sink dataset, ensuring compatibility with the AutoML task. We chose the Area Under the Curve (AUC) metric for evaluation, as it provides a robust measure of model performance, particularly in imbalanced datasets like bank churn

Week 4: MLOps, Deployment, and Final Presentation

I was responsible for developing the web application for the project. I created the front-end using HTML, CSS, and JavaScript to ensure a user-friendly interface. For the back-end, I used Flask to handle the server-side logic and to integrate the prediction model built by my colleague. The web application allows users to input customer data and receive predictions on whether a customer is likely to leave the bank. I also deployed the application on Azure to ensure it was scalable and accessible for the bank's team. My work ensured that the application was functional, visually appealing, and smoothly connected to the prediction model.