Kho dữ liệu & Hệ hỗ trợ quyết định cho bài toán Bank Customer Churn

Data Warehouse & Decision Support System

Hoa Toàn Hạc (2201917)
Mai Huy Hiệp (2211045)

Trường Đại học Bách khoa TP.HCM
Khoa Khoa học và Kỹ thuật máy tính

Tháng 11, 2025

Nội dung trình bày

Đặt vấn đề

Tại sao Customer Churn quan trọng?

- Chi phí thu hút khách hàng mới cao gấp **5-25 lần** so với giữ chân khách hàng hiện tại
- Tỷ lệ churn cao ảnh hưởng trực tiếp đến doanh thu và uy tín ngân hàng
- Cần phân tích dữ liệu để **dự đoán và ngăn chặn** churn

Thách thức

- Dữ liệu khách hàng phân tán, chưa được tổ chức tối ưu
- Thiếu công cụ phân tích đa chiều (OLAP)
- Chưa có mô hình dự đoán chính xác
- Thiếu hệ thống hỗ trợ ra quyết định

Mục tiêu đề tài

1. **Thiết kế Data Warehouse**
   - Xây dựng star schema với dimension và fact tables
   - Tối ưu hóa cho phân tích OLAP
2. **Xây dựng quy trình ETL**
   - Pipeline tự động: Extract $\rightarrow$ Transform $\rightarrow$ Load
   - Feature engineering cho phân tích và modeling
3. **Phân tích OLAP & Visualization**
   - Truy vấn đa chiều: địa lý, tuổi, thu nhập
   - Biểu đồ trực quan với Python
4. **Xây dựng mô hình ML**
   - Dự đoán khả năng churn của khách hàng
   - Đánh giá và tối ưu hóa mô hình
5. **Hệ hỗ trợ quyết định (DSS)**
   - Tích hợp DWH, OLAP, ML thành hệ thống hoàn chỉnh
   - Dashboard tương tác cho người dùng

Tổng quan dữ liệu

Nguồn dữ liệu

- **Dataset**: Bank Customer Churn Modeling
- **Nguồn**: Kaggle
- **Số lượng**: 10,000 khách hàng
- **Số biến**: 14 biến
- **Tỷ lệ churn**: 20%

Các biến chính

Nhân khẩu học:

- Age, Gender, Geography

Tài chính:

- CreditScore, Balance
- EstimatedSalary

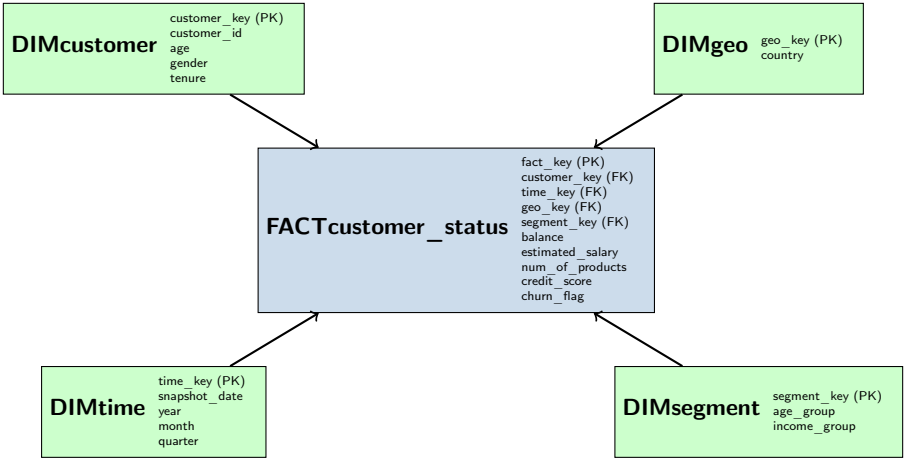
Hành vi:

- Tenure, NumOfProducts
- HasCrCard, IsActiveMember

Target:

- Exited (0=Retained, 1=Churned)

Star Schema - Tổng quan



Bảng chiều - Dimension Tables

dim_customer

Thông tin nhân khẩu học

- customer_key (PK)
- age, gender, tenure

dim_time

Thông tin thời gian

- time_key (PK)
- snapshot_date, year, month, quarter

dim_geo

Thông tin địa lý

- geo_key (PK)
- country

dim_segment

Phân khúc khách hàng

- segment_key (PK)
- age_group
- income_group

Grain của Fact Table

Định nghĩa Grain

Một bản ghi cho mỗi khách hàng
tại một thời điểm snapshot

Ý nghĩa

- Mỗi dòng trong fact table đại diện cho trạng thái của một khách hàng
- Tại một thời điểm cụ thể (snapshot date)
- Cho phép phân tích theo thời gian nếu có nhiều snapshots
- Hỗ trợ tracking sự thay đổi của customer behavior

Lợi ích của Star Schema

1. **Hiệu suất cao**
   - Truy vấn OLAP nhanh (simple joins)
   - Tối ưu cho Business Intelligence tools
2. **Dễ hiểu và sử dụng**
   - Cấu trúc đơn giản, trực quan
   - Business users dễ dàng viết queries
3. **Phân tích đa chiều**
   - Slice & Dice theo nhiều chiều
   - Drill-down, Roll-up linh hoạt
4. **Khả năng mở rộng**
   - Dễ dàng thêm dimension mới
   - Không ảnh hưởng đến hệ thống OLTP

ETL Pipeline - Tổng quan

Các bước chính

1. **Extract**: Đọc dữ liệu từ Churn_Modelling.csv (10,000 records)
2. **Transform**:
   - Data cleaning: Loại bỏ cột không cần thiết
   - Feature engineering: Tạo age_group, income_group
   - Build dimensions: 4 dimension tables
   - Build fact: Join dimensions để lấy surrogate keys
3. **Load**: Xuất 5 tables ra CSV hoặc nạp vào PostgreSQL

Feature Engineering

Age Group

```python
def categorize_age(age):
    if age < 36:
        return 'Young'
    elif age < 56:
        return 'Middle-aged'
    else:
        return 'Senior'

df['age_group'] = df['Age'].apply(
    categorize_age
)
```

Income Group

```python
def categorize_income(salary):
    if salary < 50000:
        return 'Low'
    elif salary < 100000:
        return 'Mid'
    else:
        return 'High'

df['income_group'] = df[
    'EstimatedSalary'
].apply(categorize_income)
```

Kết quả

- **dim_segment**: 9 segments (3 age groups $\times$ 3 income groups)
- Hỗ trợ phân tích theo phân khúc khách hàng

Kết quả ETL

Bảng	Số bản ghi	Mô tả
dim_customer	10,000	Thông tin khách hàng
dim_geo	3	France, Germany, Spain
dim_time	1	Snapshot date (2025-01-01)
dim_segment	9	3 age groups $\times$ 3 income groups
fact_customer_status	10,000	Measures + Foreign keys

Ưu điểm của quy trình

- **Tự động hóa**: Toàn bộ bằng Python, chạy lại dễ dàng
- **Modular**: Mỗi bước là function riêng
- **Reproducible**: Kết quả nhất quán
- **Scalable**: Dễ mở rộng cho nhiều snapshots

OLAP Queries - Ví dụ

Query 1: Tỷ lệ churn theo quốc gia

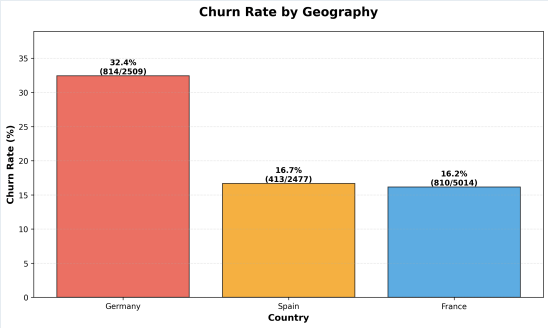
```sql
SELECT
    g.country,
    COUNT(*) AS total_customers,
    SUM(f.churn_flag) AS churned_customers,
    ROUND(AVG(f.churn_flag) * 100, 2) AS churn_rate_pct
FROM fact_customer_status f
JOIN dim_geo g ON f.geo_key = g.geo_key
GROUP BY g.country
ORDER BY churn_rate_pct DESC;
```

Kết quả

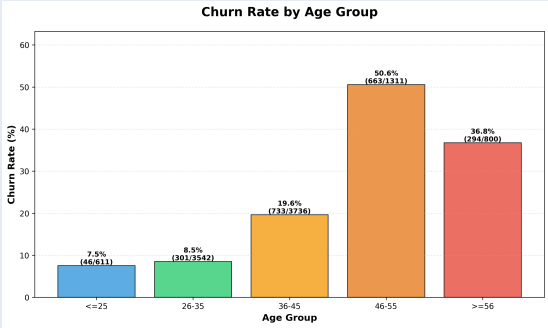
- **Germany**: 32% churn rate (cao nhất!)
- **Spain**: 17% churn rate
- **France**: 16% churn rate

Insights từ OLAP

Churn by Geography

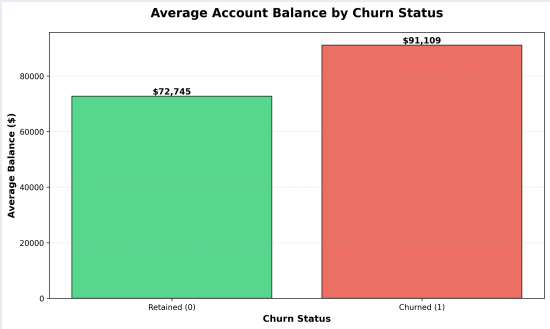


Churn by Age Group

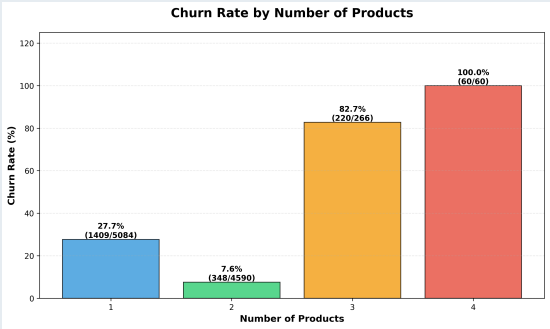


Phân tích sâu hơn

Balance by Churn Status



Churn by Products



Key Findings Surprising Insights

Key Findings

- Germany có churn rate cao gấp đôi các nước khác
- Nhóm Middle-aged (36-55) có churn rate cao nhất

Surprising Insights

- **Balance Paradox**: Churned customers có balance cao hơn!
- **Product Overload**: Khách hàng có 3-4 sản phẩm churn nhiều hơn

ML Pipeline

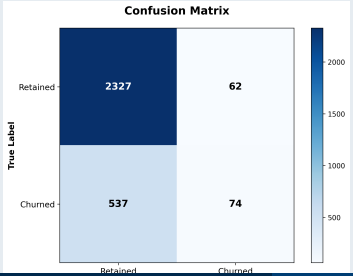
1. **Data Preparation**
   - Features: CreditScore, Age, Balance, Geography, Gender, etc.
   - Target: Exited (0=Retained, 1=Churned)
   - Train/Test split: 80/20 with stratification
2. **Preprocessing**
   - StandardScaler cho numeric features
   - OneHotEncoder cho categorical features
   - Pipeline tự động hóa toàn bộ quy trình
3. **Models**
   - Logistic Regression (baseline)
   - Random Forest (ensemble)
4. **Evaluation**
   - Accuracy, Precision, Recall, F1-Score
   - ROC-AUC, Confusion Matrix
   - Feature Importance

Kết quả mô hình

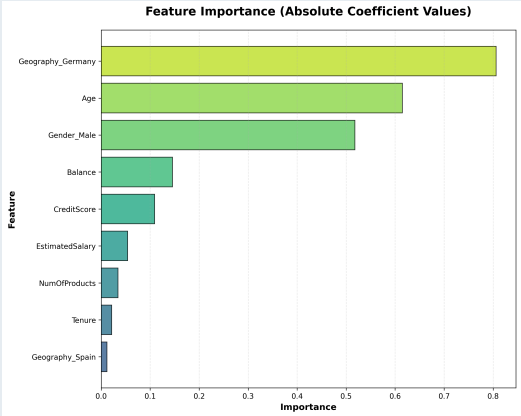
Logistic Regression

- **Accuracy**: 81%
- **ROC-AUC**: 0.85
- **Precision**: 56%
- **Recall**: 25%

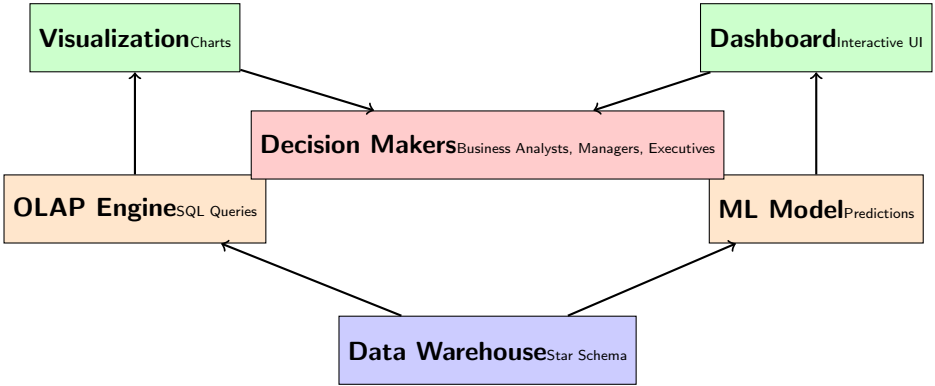
Confusion Matrix



Feature Importance



Kiến trúc DSS



Interactive Dashboard

Tính năng Dashboard

- **KPI Cards**: Total Customers, Churned, Churn Rate, Avg Balance
- **Interactive Filters**: Country, Age Group, Gender
- **6 Interactive Charts**:

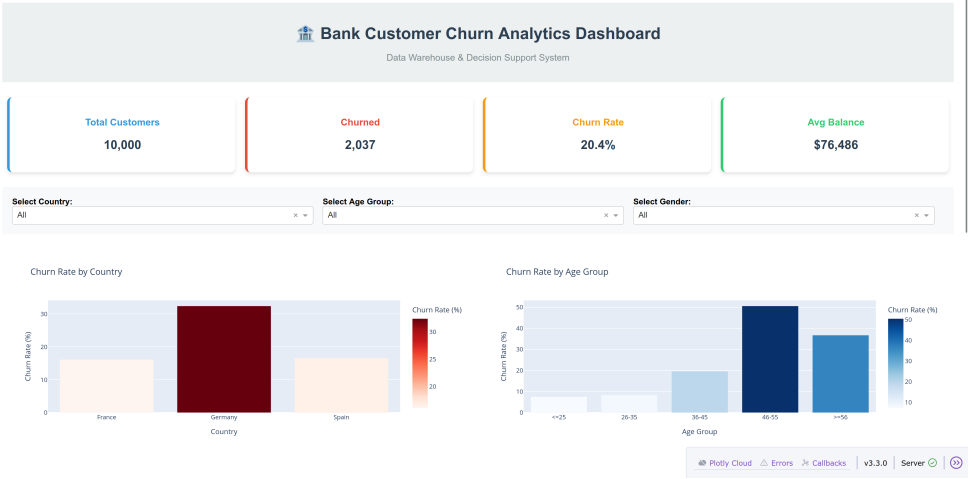
1. Churn by Country
2. Churn by Age Group
3. Balance Distribution
4. Churn by Products
5. Age Distribution
6. Churn by Tenure

- **Real-time Updates**: Charts cập nhật ngay khi thay đổi filters

Công nghệ

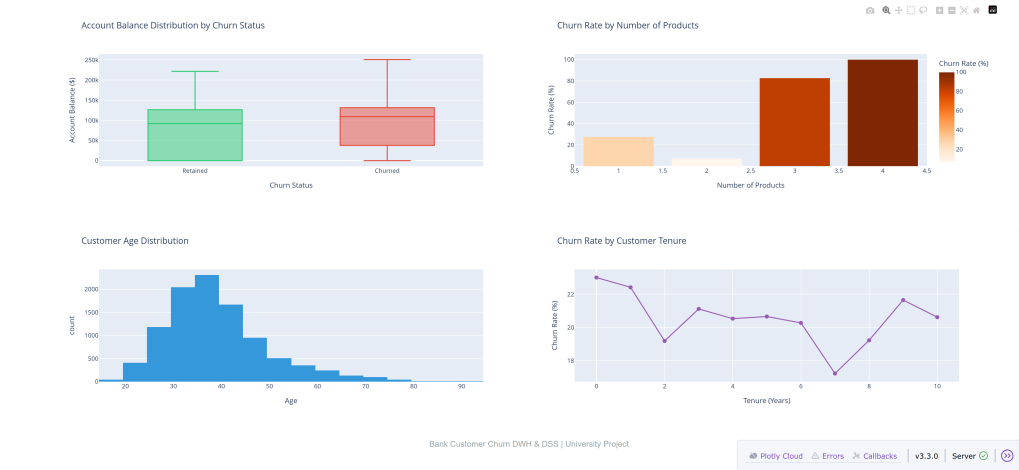
Plotly Dash - Web framework cho interactive analytics

Dashboard Screenshots



Hình: Dashboard Overview - KPI Cards, Filters, và Charts

Dashboard Screenshots (cont.)



Hình: Dashboard - Phần dưới với các charts tương tác

Ứng dụng DSS trong ra quyết định

Use Case: Phân tích churn ở Germany

1. **Bước 1**: Chọn filter Country = "Germany"
2. **Bước 2**: KPIs cập nhật → Churn rate = 32%
3. **Bước 3**: Charts hiển thị:
   - Age group 46-55 có churn cao nhất
   - Khách hàng có 3-4 sản phẩm dễ churn
   - Balance cao nhưng vẫn churn
4. **Bước 4**: **Decision** → Tập trung retention campaign vào:
   - Target: Germany, age 46-55, có nhiều sản phẩm
   - Action: Cải thiện dịch vụ, tăng lãi suất, giảm phí

Kết quả đạt được

Về mặt kỹ thuật

- ✓ Star Schema DWH (5 tables)
- ✓ ETL Pipeline tự động
- ✓ 12+ OLAP queries
- ✓ 8+ visualizations
- ✓ ML model (81% accuracy)
- ✓ Interactive Dashboard

Về mặt nghiệp vụ

- ✓ Xác định yếu tố chính ảnh hưởng churn
- ✓ Phân khúc khách hàng
- ✓ Dự đoán churn proactive
- ✓ Đề xuất chiến lược retention
- ✓ Hệ thống DSS hoàn chỉnh

Key Insights

- Germany có churn rate cao nhất (32%)
- Khách hàng 46-55 tuổi dễ churn nhất
- Khách hàng có 3-4 sản phẩm có nguy cơ cao
- Balance cao không đảm bảo retention

Hạn chế & Hướng phát triển

Hạn chế

- Dữ liệu tĩnh (1 snapshot)
- Mô hình ML còn đơn giản
- Recall cho Churned thấp (25%)
- Chưa deploy production
- Thiếu A/B testing

Hướng phát triển

- Thu thập time-series data
- Thử XGBoost, Neural Networks
- Xử lý imbalanced data (SMOTE)
- Deploy lên cloud (AWS, GCP)
- Tích hợp CRM, email marketing
- Automated alerts & actions
- CLV prediction
- Next-best-action engine

Tech Stack

Công nghệ sử dụng

- **Data Warehouse**: Star Schema (CSV/PostgreSQL)
- **ETL**: Python (pandas, numpy)
- **Visualization**: matplotlib, plotly
- **Machine Learning**: scikit-learn
- **Dashboard**: Plotly Dash
- **Database**: PostgreSQL (optional)
- **Notebooks**: Jupyter

Project Structure

- `data/`: raw, interim, processed
- `src/`: data, models, visualization
- `sql/`: DDL, OLAP queries
- `reports/`: LaTeX report + figures

Kết luận

Tổng kết

Đề tài đã thành công xây dựng một hệ thống **Data Warehouse Decision Support System** hoàn chỉnh cho bài toán Bank Customer Churn, bao gồm:

- Kho dữ liệu tối ưu với star schema
- Quy trình ETL tự động và reproducible
- Phân tích OLAP đa chiều với insights giá trị
- Mô hình ML dự đoán churn chính xác
- Dashboard tương tác cho decision makers

Kết luận

Giá trị thực tiễn

Hệ thống không chỉ là bài tập học thuật mà còn có thể triển khai thực tế, giúp ngân hàng:

- Giảm tỷ lệ churn
- Tăng customer lifetime value
- Ra quyết định dựa trên dữ liệu (data-driven)

DEMO

Interactive Dashboard

`http://localhost:8050`

Q & A

Cảm ơn quý thầy cô đã lắng nghe!

Tài liệu tham khảo

- Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3rd ed.). Wiley.
- Kaggle. *Bank Customer Churn Modeling*. `https://www.kaggle.com/datasets/shrutimechlearn/churn-modelling`
- Pedregosa, F., et al. (2011). *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.
- Plotly Technologies Inc. (2015). *Collaborative data science*. `https://plot.ly`

Backup: SQL DDL Example

```sql
CREATE TABLE dim_customer (
    customer_key SERIAL PRIMARY KEY,
    customer_id INTEGER NOT NULL,
    age INTEGER,
    gender VARCHAR(10),
    tenure INTEGER
);

CREATE TABLE fact_customer_status (
    fact_key SERIAL PRIMARY KEY,
    customer_key INTEGER REFERENCES dim_customer(customer_key),
    time_key INTEGER REFERENCES dim_time(time_key),
    geo_key INTEGER REFERENCES dim_geo(geo_key),
    segment_key INTEGER REFERENCES dim_segment(segment_key),
    balance DECIMAL(15,2),
    estimated_salary DECIMAL(15,2),
    num_of_products INTEGER,
    credit_score INTEGER,
    has_credit_card INTEGER,
    is_active_member INTEGER,
    churn_flag INTEGER
);
```

Backup: Python ETL Code

```python
import pandas as pd

# Extract
df = pd.read_csv('data/raw/Churn_Modelling.csv')

# Transform - Feature Engineering
def categorize_age(age):
    if age < 36: return 'Young'
    elif age < 56: return 'Middle-aged'
    else: return 'Senior'

df['age_group'] = df['Age'].apply(categorize_age)

# Build dimension tables
dim_customer = df[['CustomerId', 'Age', 'Gender', 'Tenure']].copy()
dim_customer = dim_customer.drop_duplicates(subset=['CustomerId'])
dim_customer.reset_index(drop=True, inplace=True)
dim_customer['customer_key'] = dim_customer.index + 1

# Load
dim_customer.to_csv('data/processed/dim_customer.csv', index=False)
```