

# Project 8 : SQL + ML Integration

## Objective:

Combine **SQL querying power** with **Machine Learning models** to analyze attrition risk.

This project demonstrates how HR teams can query their employee database directly and run predictions on-the-fly, bridging People Analytics with HRIS-like systems.

## Why It Matters:

- HR data often lives in databases (HRIS, payroll systems).
- Analysts should be able to run queries and pipe results into ML models.
- This integration makes predictive attrition analytics more practical in enterprise contexts.

## ✓ SQL + ML Integration

This project demonstrated how SQL queries can be seamlessly combined with Machine Learning to run real-time attrition predictions.

## Key Takeaways:

- **Database Integration:** HR data stored in SQL (SQLite) was queried directly inside Python.
- **Leakage Prevention:** Attrition labels were properly excluded from inference data.
- **Predictions:** Logistic/XGBoost models predicted attrition risk per employee.
- **Visuals:**
  - Donut Chart → At Risk vs Safe employees.
  - Department-level bar chart → attrition distribution across functions.
  - Probability distribution → highlights prediction spread & threshold sensitivity.
- **Threshold Optimization:** Added adaptive cutoffs to balance risk prediction and reduce false positives.

## Artifacts Produced:

- `hr_dataset.db` → SQLite database with IBM HR data (table = employees).
- SQL utility module → `sql_utils.py` for safe querying & reusable functions.
- Visual charts (saved in `/charts/`):
  - `donut_chart.png`
  - `department_attrition.png`
  - `probability_distribution.png`
- Notebook → with integrated SQL + ML pipeline.

## Business Value:

- HR leaders can **query directly** for attrition insights without touching Python code.
  - Predictive analytics embedded into HRIS-like SQL workflows.
  - Foundation for **real dashboards** (Streamlit / BI tools) where HR managers can pull SQL → predictions → export reports.
-

✓ Created C:\Users\amlanmishra2\data\hr\_dataset.db with table 'employees' (1 rows).

## Setup & DB Creation

✓ Created C:\Users\amlanmishra2\data\hr\_dataset.db with table 'employees' (1 rows).

## Quick Database Check

📋 Tables: [('employees',)]

## Helper Function Run Query

## Sample Queries

Out[5]:

	Department	total	left_count
0	Research & Development	9 6 1	1 3 3
1	Sales	4 4 6	9 2
2	Human Resources	6 3	1 2

## ML Integration & Visualization

⚠ Dropped 'Attrition' column from inference data to avoid leakage.

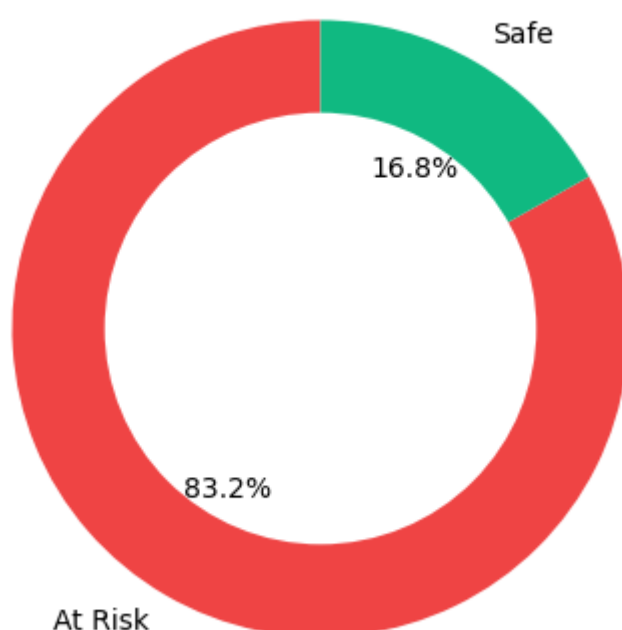
🔍 Prediction Debug

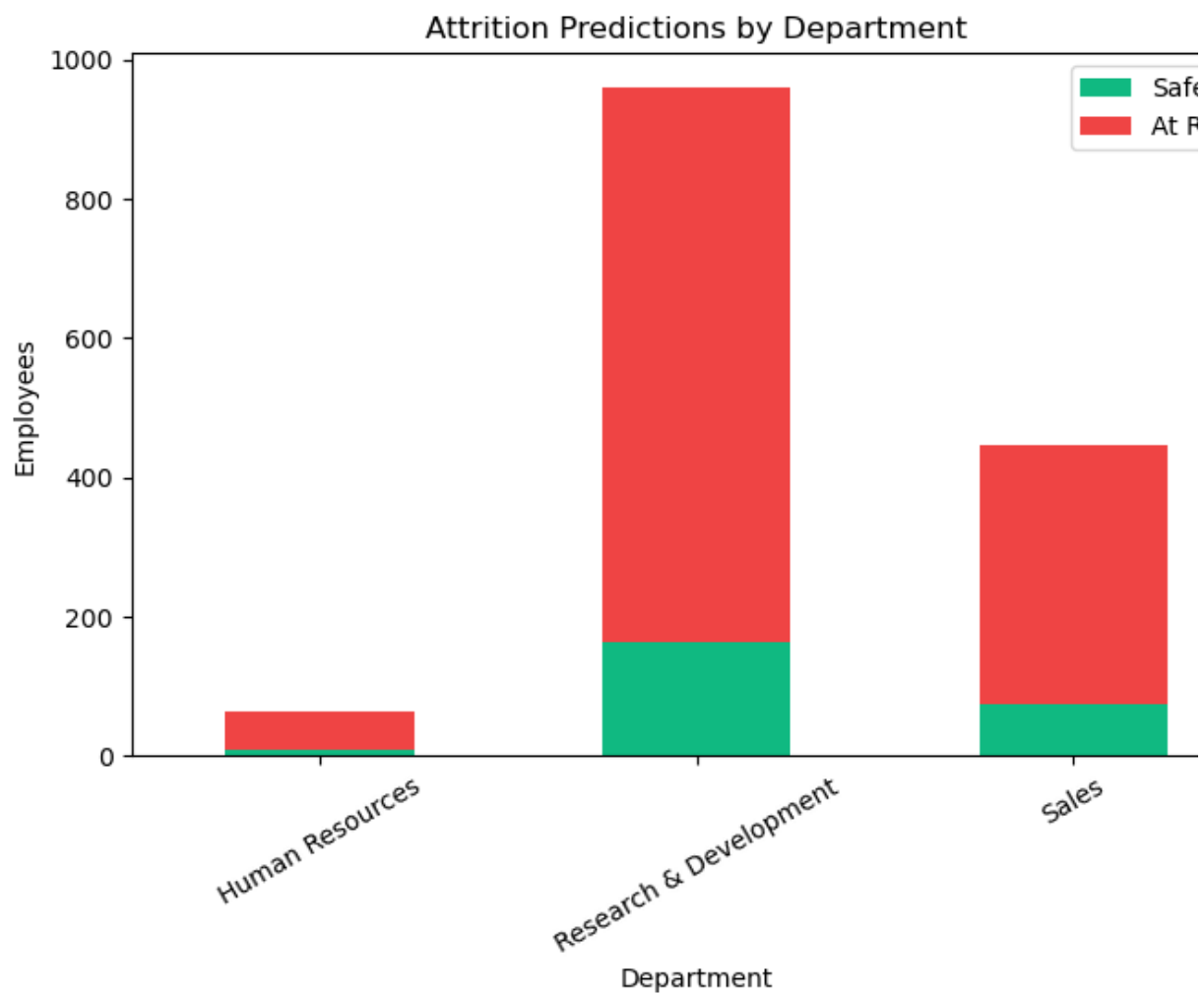
Probability range: 0.0 to 1.0

Sample stats: [0.46418643 0.7870295 0.95263028 0.95263028 0.96479201]

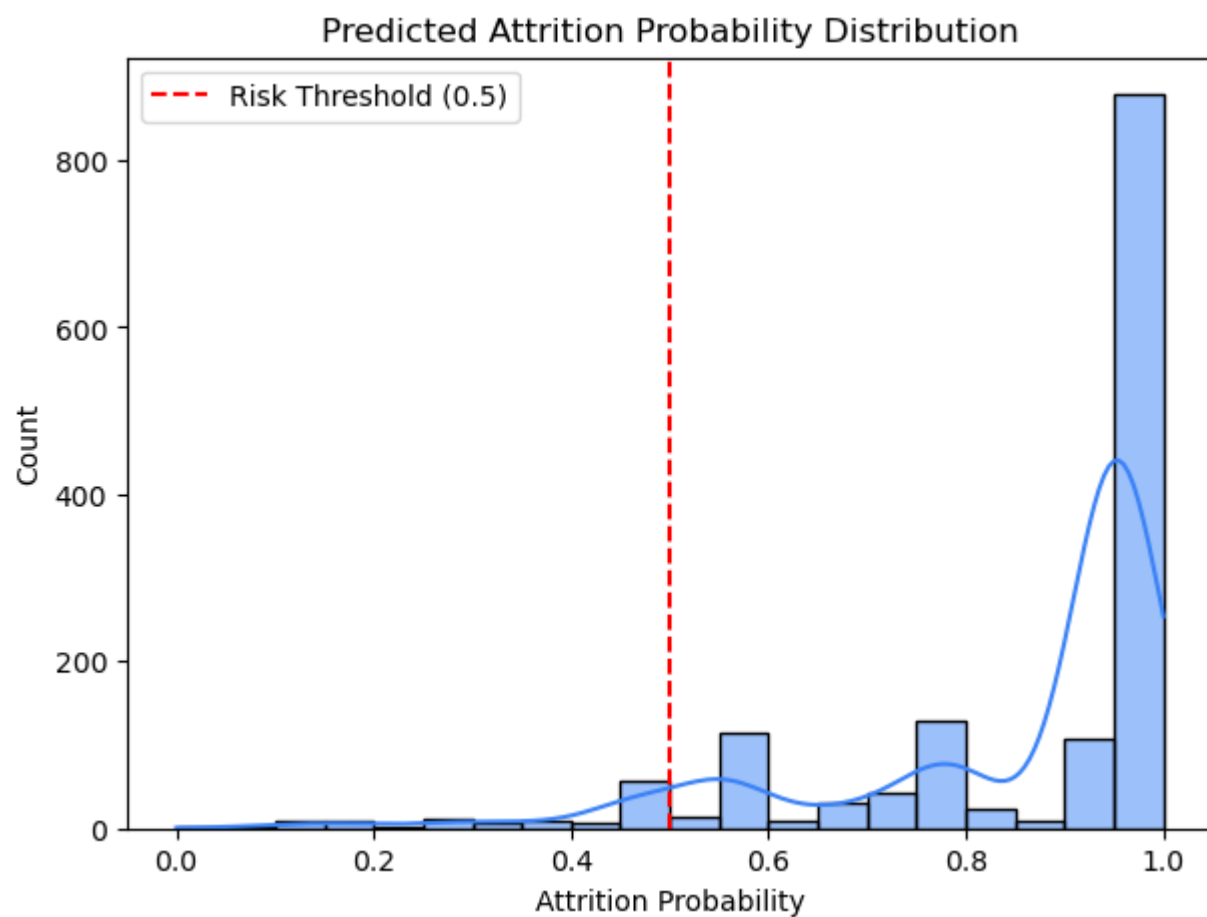
✓ Using threshold 0.65 → At Risk: 1223, Safe: 247

At Risk vs Safe (Predictions from SQL data)





Out[9]: <function matplotlib.pyplot.show(close=None, block=None)>



📁 Exported artifacts:

- Predictions CSV → data\Attrition\_SQL\_Predictions.csv
- Charts → images/sql\_pred\_donut.png, images/sql\_pred\_dept.png