Name: Cushing, Bradley
Date: 11/04/24
NYU ID: N10695516
Course Section Number: CSCI-GA-2433-001
Project Part #2 Report

Files included

- cushing_p2_su24_report.pdf
- cushing_p2_su24_logical_schema.sql
- cushing_p2_su24_cb_er_model.sql
- cushing_p2_su24_dol_title_wages.csv
- cushing_p2_su24_warranties_actual.csv

Logical database schema

The logical database schema has been created in a file cushing_p2_su24_logical_schema.sql. This includes all the necessary SQL code (DDL) to create the database according to the relational schema specified in the first step.

I've also updated the conceptual schema first to include PERSONAL_INFORMATION for users which is described below for those that want to submit it and two new Discount_pct and Price fields in the WARRANTY table which uses the price set by the prediction explained below. I've included a new copy of the conceptual schema in cushing_p2_su24_cb_er_model.sql which also reflects this change. We expect the Discount_pct and Price to be set based on what exists at time of record creation when the prediction will be made.

Unstructured data collection

- cushing_p2_su24_dol_title_wages.csv
- cushing_p2_su24_synthetic_external_risk.csv

There are two sets of unstructured data I've "collected" which are external to the company. The first set of data that I felt would be useful from the Department of Labor. The idea is that based on wage information for specific job titles I would be able to predict and benchmark salaries paid for employees within the Circuit Blocks Inc. organization. I've included the data set called cushing_p2_su24_dol_title_wages.csv which includes this data

Secondly, I've also decided to generate some artificial, external data to use which represents anonymous personal information of people that would be collected outside of the organization. We could imagine this data as coming from Census.gov or another external government organization. This data also comes with an indication of whether a certain profile was deemed risky by the government in some way. Maybe they were late on filing taxes or owe taxes which haven't been paid. Many scenarios are possible but we consider this risk indicator good for us to

map to risk related to warranties inside of our own business. We'll be using this artificial data to train a Logistic Regression model and predict the risk inside of our business for users who submit their personal information to get a warranty discount. The reason I'm using synthetic data is because it was easier to use for prediction using Logistic Regression.

I searched online quite extensively for good external data that I could use to train a Logistic Regression model but I had a very hard time finding something that would be good for any use case in my business. Therefore, after discussing with the professor I chose to create my own dataset that could be used for training such a model and make a simple prediction. I picked attributes that I thought could be interesting indicators of whether a person might be participating in risky behaviour or have unpredictability around them.

The general idea

The premise is that we would allow customers who have purchased the product to give us additional personal information about themselves and their household and we would offer them a discount in return. The company Circuit Blocks Inc. sells warranties for the products that they make and sell to customers. Warranties are order specific and are priced at 20% of the original product order price. Anyone who submits additional, personal information will only have to pay 15% of the order price (we'll use this information to inform marketing campaigns which is why we value it) and those we predict as not risky will only have to pay 10%. When someone uses a warranty they get a full replacement for that particular order.

Price per warranty

- (submits no information) 20% of original order
- (submits information but risky) 15% of original order
- (submits information but not risky) 20% of original order

Semi-structured data set

Risk categorization is based on someone's living situation. Please see the synthetic data in cushing_p2_su24_external_risk_actual.csv for what will be used to train a prediction model. I've included a few rows from the sample data below to give an overview of its contents.

Example CSV data

Age	Kids_count	Pets_count	Siblings_count	Income	Has_risk
35	5	3	0	97638	1
61	3	5	2	87510	1
56	1	1	4	173967	0

How insights inform EDA created

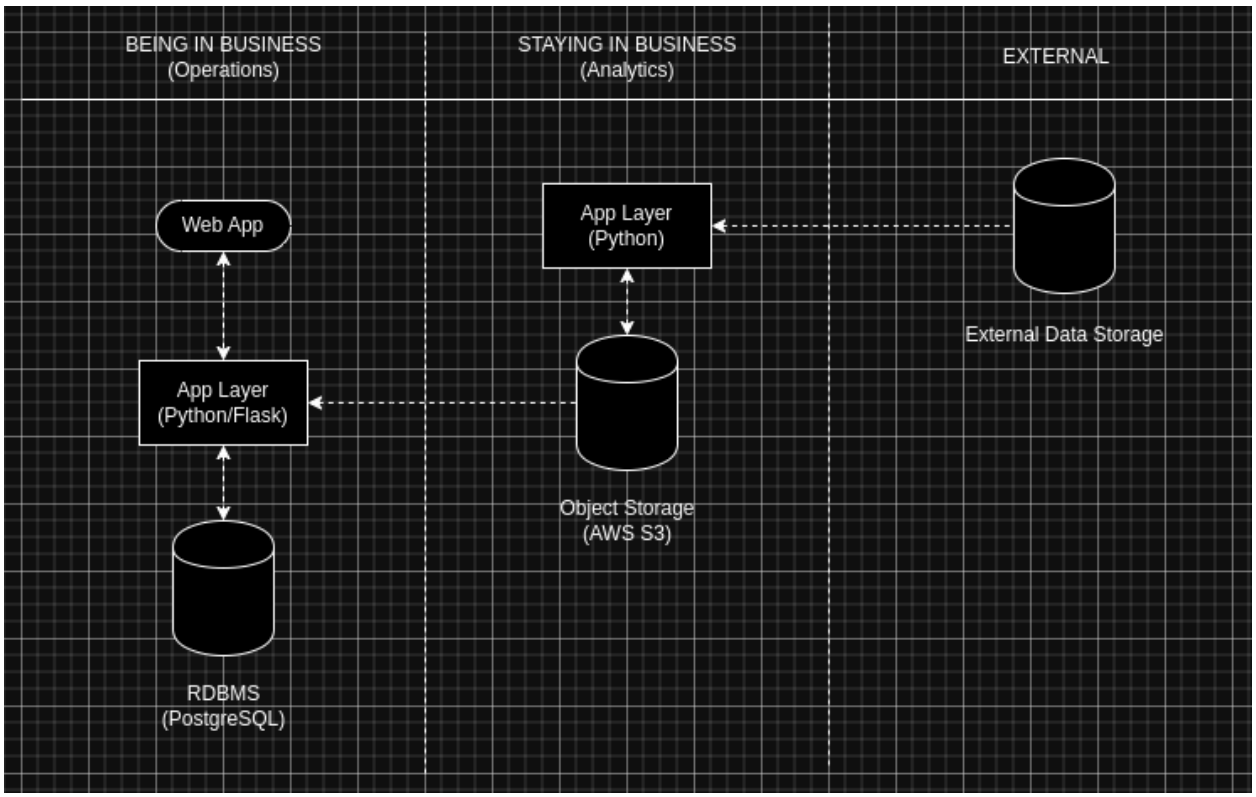
The original data contains an indication of whether a person is risky. In training then, the first 5 columns of data are used as an indicator for whether the person purchasing a warranty for the product should be considered as “risky” or not to us. We will process this information submitted by a user who has purchased the product if they choose to also purchase a warranty.

Semi-structured files

We consider unstructured data to be semi-structured in reality as what we want to process from external sources will typically be in CSV, XML, or JSON which we can read, parse, normalize, and reformat into something structured like a table in our relational database. Then we can query our relational database with this structured data to gain insights and inform the business. We won’t consider completely unstructured data like text documents, audio files, etc.

Application components

Because our conceptual and logical schemas are for relational databases only we only model the relational database using the conceptual and logical models. Below we list all components that make up the entire application including operations, analytics, and external data sources which we can imagine provides the data we use to train our prediction model.



Being in Business

This contains the main application components which make up the user interface (web application), application layer (Python/Flask server), and RDBMS (PostgreSQL). This is the set of components that serve the operational side of the business. An example use case here would be a customer adds their personal information to the database for a warranty, but in order to determine the risk of the customer which informs the price of the warranty, the application layer fetches the latest model from the S3 object storage on the analytics side.

Staying in Business

The analytics side is where we train our model from the external data and store the trained model in the S3 object storage. We can imagine a recurring job which takes the external data and creates a CSV file in S3 at some interval. When this is done we also have a job which trains the model using the latest external data and saves this model to S3 as well. This latest model will be the one that is always retrieved from the operations side when needing to inference.

External data source

This is just a publicly available data source that is accessible either by scraping their website or which provides API access which we can use to retrieve updated data for training.

Connecting to a use case

Here we include the full flow for allowing a user to enter information into a form to purchase a warranty, the making of a prediction using our model and entered information, showing this information to the user, and persisting this information in the relational database.

- Assume prediction model for risk is already trained
- User places an order to purchase some product
- User decides to purchase a warranty for their product
- User is given option to submit personal information for a discount
- User enters the OrderId they would like to purchase a warranty for
- User inputs the 5 requested fields of personal information
- Application layer fetches the latest model from analytics database
- Application layer runs prediction model to determine associated risk
- Given the risk prediction we then notify the user of the price
- If the user agrees we enter the warranty and price in the database

Normalization

I started populating a local PostgreSQL database with mock data to start thinking about use cases which helped me to realize where I could optimize and normalize things. Below are some of the changes I made to the E-R model and DDL as a result of normalization. A lot of relationships have been flipped to make sure we don't create NULL values in any of the tables where we don't absolutely need to. Some new tables were created to make relationships better to help achieve higher normal forms as well.

Entity changes

- New INVENTORY_BACKORDER table was created for the M:N relationship between INVENTORY and BACKORDER. This better models this relationship and removes NULL values that were present in Inventory since most of Inventory won't be on backorder. This involved removing the boolean is_backorder flag which we now no longer need. We also added a status attribute on Backorder to state whether it's "open" or "filled".
- Both INVENTORY and BACKORDER entities were slightly modified
- PAYROLL is now related to a new EMPLOYEE entity we've created which models inheritance. This was needed to simplify the relationship between employees and payroll. Before we had specific types of employees, which were DESIGNER and PROGRAMMER which each had their own Id attribute and were independently related to payroll which led to unwanted NULL values in payroll because you could only be one type of employee. We've also made use of the Id from employees in other tables where employee Ids are needed like the FILES table.
- We flipped the relationship between CREDIT_CARD and CUSTOMER because every credit card has a customer, meaning the customer Id is now a foreign key in the credit card table. This also removes NULL values we would have encountered otherwise. The same thing was done for PERSONAL_INFORMATION and CUSTOMER to avoid NULL values. MEMBER and CUSTOMER was the last relationship flipped here. Now the customer table is much cleaner and will have no NULL values. Customer Id is also the primary key for the member table which no longer has a different Id.
- The relationship was flipped for Warranty and Order for the same reasons
- We've also now made it so a Customer IS_A Prospect. You don't have to be a Customer to be a Prospect but all Customers are automatically added to this group so they can receive emails and other types of promotions. We've left Email and Name fields in prospect open and not restricted to those in Customer for that reason.

Relationship changes

- Added Product SOLD_BY Order
- Removed Order SELLS Product
- Added Inventory HAS_MANY Inventory_Backorder
- Added Backorder HAS_MANY Inventory_Backorder
- Added Designer IS_A Employee
- Added Programmer IS_A Employee
- Added Payroll PAYS Employee
- Removed Payroll PAYS Designer
- Removed Payroll PAYS Programmer
- Added Credit card STORED_FOR Customer
- Removed Customer STORES Credit Card
- Added Personal information GIVEN_BY Customer
- Removed Customer GIVES Personal information
- Added Member IS_A Customer
- Removed Customer BECOMES Member

- Added Warranty REGISTERED_FOR Order
- Remove Order REGISTERS Warranty
- Added Customer IS_A Prospect
- Removed Prospect BECOMES Customer