

Background

This Python program is designed to help Dillard's business users change their operating model in a way that allows them to turn data into business insights, better connect with customers, and more efficiently operate their stores (both brick-and-mortar and ecommerce). This prototype has been designed to be a platform interface for two different segments of Dillard's stores, sales and customer service.

Sales/Marketing Portal Concept

Solution Design

This program will run modules Pretty Table and sqlite3. The module Pretty Table is a simple python package designed to represent data in a visually appealing way and easy to use for business users. It is designed to display output data in a table format rather than having to struggle with lists that are hard to read or interpret. The good thing about PrettyTable is that the user can control the width of columns, the alignment of the text results, creating a border, sorting, and much more. This can be easily adjusted per the users' request.

Again, you will need the SQL database that users need to run reports. A relational database management system (RDBMS) usually requires a separate server process to function and require a TCP/IP protocol for the requests. However, the SQLite python package allows database files to become readable and manipulation of the file without requiring a separate server process. It is "lite" or light weight in terms of setup or required resources. This library is integrated with Python to access the database and interact to read and write directly from the database.

How It Works

This program is great because end-users that have no experience in pulling information from a database by using SQL will have a way to view the data with little to no effort. The program starts with

a welcome message, and then the user is asked to specify which portal they are looking to access (Marketing, Inventory, Customer Service, or the twitter feed).

For the program to work, the user will need to install PrettyTable in their system and have access to the database. Once the program is running it will prompt the user for input. All the user needs to do is follow the instructions.

Customer Service Portal

Concept

This phase of the prototype was designed to allow customer service representatives to monitor, via text mining, what customers are saying about their recent shopping experience. Faster access to this feedback allows a customer service representative to personally reach out to the disgruntled customer in an attempt to mitigate the issue. Users can open up an automated search via Python and Tweepy, that mines data in real time, and then provides the results in the requested format with the requested information, in a consolidated view or table. A marketing team can also use this tool (with some increased functionality to be added in phase 2) in order to analyze not only what is being said about their company, but how many people are talking about them, or how many times a tweet is being retweeted/favorited. This awareness allows for the promotion of positive publicity.

Solution Design

Tweepy uses the twitter api in order to handle authentication, connection, creating the session, and reading incoming messages. Tweepy provides a much simpler, and neater route for accessing twitter feeds, as opposed to parsing Twitter. StreamListener, a function of Tweepy, then streams the results of the search onto the desktop of a customer service representative. This turns your computer screen into a live twitter feed! The other key module used in this design is “json” due to the fact that the Tweepy

dumps the entire block of JSON information for each of the tweets that are streamed as a result of the search query.

How it Works

After importing tweepy, you then will want to import StreamListener, OAuthHandler, and Stream (all from within tweepy). Authentication for access to the twitter API is handled by OAuthHandler within Tweepy, which makes this process much simpler.

Once the user has accessed the customer service portal and tweeter feed, they will be prompted to answer a request in regards to what they are looking for. Simply input a keyword, (likely Dillards), and then Tweepy will start scanning Twitter in real time as items are posted. As tweets are found containing the keyword, it is then streamed onto the users monitor in a format that can be easily read and interpreted, and then saved to a csv.

The csv is simply for ease of use within the demo. This will be modified upon implementation into your environment so that the results are loaded into a database of your choice. At this point, we will be able to add functionality to the customer service portal which allows them to access this database, via this platform, in order to analyze trends in customer service issues.

Notice

Please note, because this is a stream of live twitter feeds in real time, the results will continue to come in until manually exited. Ideally, this would be running all day on the screen of the marketing and customer service department to detect trends and/or respond in real time to issues. Once the user logs out of the platform, the streams will cease.

Dillard's Enterprise Menu

Appendix

Figure A1 – Dillard's Database Entity Relationship Diagram

Dillard's Database

Entity Relationship Diagram: ua_dillards_2016

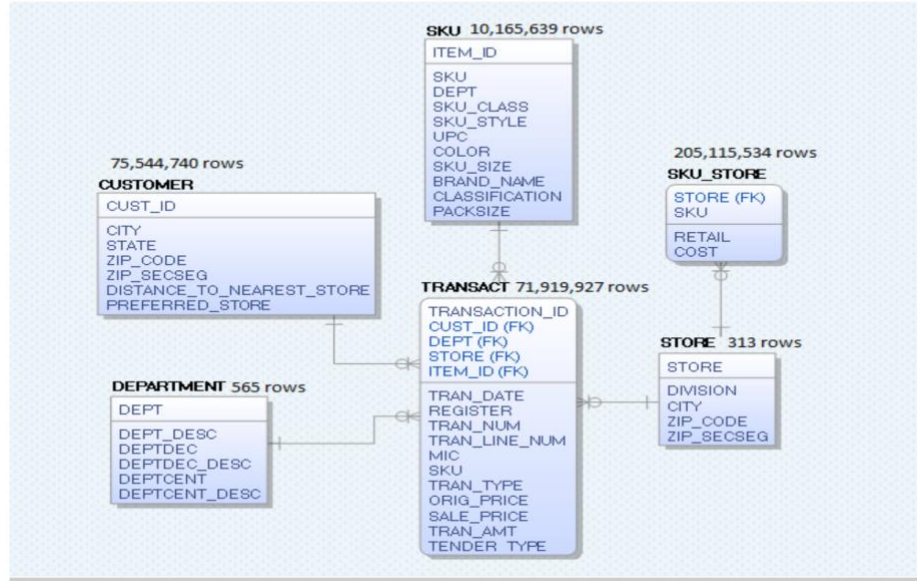


Figure A2 – Data Dictionary

Data Dictionary

Department:

Abbreviation	Term Name	Short Description	Long Description
DEPTCENT	Century	The first digit of a department code, a way to classify departments at a higher level	Values include 00XX (Cosmetics), 01XX (Ready-to-Wear), 02XX (Lingerie/Accessories), 03XX (Juniors), 04XX (Childrens), 05XX (Mens), 06XX (Shoes), 07XX (Home Decor), and 08XX (Big Ticket).
DEPTCENT_DESC	Century Description	The descriptive name of a century.	Display values (and their corresponding Century values) include COSMETICS (00XX), READY-TO-WEAR (01XX), LING/ACCESS (02XX), JUNIORS (03XX), CHILDRENS (04XX), MENS (05XX), SHOES (06XX), DECOR. HOME (07XX), and BIG TICKET (08XX).
DEPT_DESC	Department Description	The name for a department collection of merchandise within a store format.	Some departments are considered classification departments (Dept 0234) TAILORED FASHION, (Dept 0254) Belts While other departments are Brand Departments (Dept 0262) Spanx, (Dept 0024) Kiehls
DEPT	Department	The Dillard's unique identifier for a collection of merchandise within a store format.	Possible values include 0012, 0123, 0244, 0323, 0456, 0533, 0686, 0711, 0844
DEPTDEC_DESC	Decade Description	Descriptive name representing the decade to which a department belongs.	Possible values include (056X) Mens Pants, (026X) Hosiery, (004X) Mens Cosmetics
DEPTDEC	Decade	The first and second digit of a department code, a way to classify departments at a higher level.	Possible values include 056X, 026X, 004X

Dillard's Enterprise Menu

SKU_STORE:

Abbreviation	Term Name	Short Description
SKU	Stock Keeping Unit	Dillard's assigned number that identifies an item by size within a color of a style for a vendor.
STORE	Store	The numerical identifier for any type of Dillard's location.
COST	Cost	The price charged by a vendor for a product
RETAIL	Current Retail	The current price of an item

STORE: Store 698 (Maumelle) services the online orders

Abbreviation	Term Name	Short Description
STORE	Store	The numerical identifier for any type of Dillard's location.
DIVISION	Division	The division to which a location is assigned for operational purposes.
CITY	City	The city name of a location address.
STATE	State	The state abbreviation of a location address.
ZIP_CODE	Zip Code	The 5 digit zip code of a location address
ZIP_SECG	Zip_Secseg	The 4 digit code of a neighborhood within a specific zip code

TRANSACTION:

Abbreviation	Term Name	Short Description	Long Description
TRANS_DATE	Transaction Date	Calendar date the transaction occurred in a store	
STORE	Store	The numerical identifier for any type of Dillard's location.	
REGISTER	Register	Device used to ring sales	
TRANS_NUM	Transaction Number	Sequential number of transactions rang on a register	
TRANS_TIME	Transaction Time	Time of day the transaction occurred	
CUST_ID	Customer Identifier	Surrogate key created and maintained by the data warehouse representing a unique instance of a customer.	An individual may actually exist in the database more than once but with different attributes. For example the first name may be abbreviated or the last name may be different, etc.
TRANS_LINE_NUM	Transaction Line Number	Sequential number of each element of a transaction	A trans_line_number is incremented for each element of a transaction or receipt. Each SKU rang is assigned a unique line number, sales tax is also a line item
DEPT	Department	The Dillard's unique identifier for a collection of merchandise within a store format.	
MIC	Manufacturer Identification Code	Manufacturer Identification Code used to uniquely identify a vendor or brand within a department.	
SKU	Stock Keeping Unit	Dillard's assigned number that identifies an item by size within a color of a style for a vendor.	
QTY	Quantity	The number of a specific SKU	
TRANS_TYPE	Transaction Type	An identifier for a 'P'urchase or 'R'eturn type of transaction or line item	
TRANS_AMT	Transaction Amount	The transaction total the customer paid for the merchandise	Trans Amt includes sales tax, delivery fees and any other line item required by the transaction
TENDER_TYPE	Tender Type	The specific instrument the customer used to complete the transaction	Cash, check, credit, gift cards, return

CUSTOMER:

Abbreviation	Term Name	Short Description	Long Description
CUST_ID	Customer Identifier	Surrogate key created and maintained by the data warehouse representing a unique instance of a customer.	An individual may actually exist in the database more than once but with different attributes. For example the first name may be abbreviated or the last name may be different, etc.
CITY	Household City	City where the customer lives	
STATE	Household State	State where the customer lives	
ZIP_CODE	Household Zip Code	Customer's zip code	
ZIP_SECG	Household Zip Plus 4	4 digit number identifying a neighborhood for the customer	
DISTANCE_TO_NEAREST_STORE	Household Distance to Nearest Store	Miles from customer's house to the closest Dillard's store	
PREFERRED_STORE	Household Preferred Location Id	Dillard's store number the customer prefers to shop in regardless of distance to customer's address	