D211: Advanced Data Acquisition

Bailey Psuik

Western Governors University

January 2024

Part I. Data Dashboards

A. Data dashboards

Attached to this report's submission is a Tableau workbook containing a storyboard composed of four dashboards that support executive decision-making at WGU Telecom through use of data visualizations.

A1. Data sets

Data set #1: WGU churn data

- This data set contains information on WGU Telecom's customers such as customer demographic information, monthly charges, tenure length, and more. This data set was provided pre-loaded into the PostgreSQL database in the Labs on Demand (LOD) environment.

Data set #2: External competitor data

- This data set from a "competitor" telecommunications company was accessed from Kaggle and contains similar observations to what WGU Telecom has collected about its customers.

A2. Dashboard installation

In order to install the dashboard, users will need the following downloaded (which are provided as attachments with this report):

- Tableau packaged workbook: Psuik_D211 Dashboards - Packaged WB.twbx
- External data set: D211_WGU_ChurnDataset.csv
- SQL query used to combine the data sets: D211_SQL_Query

Step 1: From the LOD home screen, open the File Explorer. Navigate to where the Tableau workbook was downloaded and open the file.

Step 2: A dialogue box will prompt the user to enter credentials for the localhost server on PostgreSQL. LOD provides these credentials, and they are below for reference:

Username: postgres
Password: Passw0rd!

Enter these credentials and sign in.

Step 3: The dashboards will appear in Tableau for viewing and interactivity.*

*Troubleshooting note: If the data visualizations do not populate once Tableau opens, the user will need to execute the SQL query in PostgreSQL to combine the data. If this is the case, follow the steps below:
Step 1: Open pgAdmin4 on the LOD desktop. If a blank window appears, simply click on the pgAdmin icon in the bottom toolbar.

Step 2: In the panel on the left-hand side, navigate to Servers > churn > Tables

Step 3: Right-click on Tables. Select the Query Tool.

Step 4: In the blank query tab that appears, copy and paste the SQL script from the attached file (D211_SQL_Query), and run the code to populate the necessary tables, and save.

Step 5: Open Tableau Public. In the left-hand panel, under Connect, navigate to “To a Server”. Click “More…”. Select PostgreSQL. Enter the PostgreSQL credentials provided in the LOD environment and sign in.

Step 6: A new workbook will open. Under the Connections panel the localhost for PostgreSQL will be listed along with the database name and the tables within it.

Step 7: Open the packaged Tableau workbook attached with this report. A dialogue box will prompt the user to enter credentials for the localhost server on PostgreSQL. LOD provides these credentials, and they are below for reference:

Username: postgres
Password: Passw0rd!

Enter these credentials and sign in.

Step 8: The dashboards will now appear in Tableau for viewing and interactivity.

A3. Dashboard navigation

The workbook contains four dashboards, on four pages, that comprise a storyboard displaying various comparisons between data from WGU Telecom and data from its rival telecommunications company. Users can view the different pages by selecting the page name at the top, or by clicking the left and right arrows adjacent to the page names. Each page is outlined below, along with further navigation instructions where applicable.

Page 1: Churn Rates

Churn rates are calculated for both WGU Telecom and its competitor, and are demonstrated by pie charts. Below this pair of pie charts is an additional pair of pie charts where users can select different combinations of demographic data to display and compare churn rates for each provider using these filters.

Page 2: Household Member Data

Average customer tenure and average monthly churn are represented here, respectively, broken out by members in customer households. Users can filter for ‘All’ to compare values for each provider side by side, or filter by individual provider to view rates for their customers. A color scale is provided for each chart, with darker colors representing higher values.

Page 3: Contract Data

Contract types are broken out here both by frequency and average tenure of customers with these contract types. The top graph represents the count of customers with each contract type with their respective provider. The bottom graph represents the average tenure, in months, of customers with each contract type with their respective provider.

Page 4: Tenure & Charge

Customers’ monthly charges and average tenure lengths are represented on this page. At the top are two horizontal bar graphs representing both the average monthly charge per provider, and the average monthly charge per provider based on whether or not their customers churned. Below these are two additional horizontal bar graphs displaying both the average tenure per provider and the average tenure per provider whether or not their customers churned.

A4. SQL code

The complete SQL code used to prepare and clean the data is provided both in the attached text file and in the appendix of this report.

Part II. Demonstration

B. Panopto presentation

A Panopto presentation detailing the dashboard creation and walkthrough of the storyboard is accessible via this link.

Part III. Report

C. Written report

The report below outlines the data exploration, use of SQL code, and analysis of the combined data.

C1. Dashboard alignment

Executive level decision-makers for WGU Telecom are interested in improving customer recruitment and retention, as well as launching new product features. The team is interested in key customer characteristics, behavior and demographics. The purpose of the dashboard is to uncover insights about WGU Telecom's customers, in addition to finding areas that can allow them to improve customer retention, recruit more customers, and offer improved products for these customers. By analyzing internal data from WGU Telecom and comparing it to similar data collected by a rival company, the dashboard provides the team with insights into this data, and how to reach their goals.

C2. Business intelligence tool

The business intelligence tool selected for this analysis is Tableau, specifically Tableau Desktop Public Edition. Tableau is a widely-used business intelligence tool known for its ease of use in building visualizations for large data sets, and sharing findings with audiences effectively. In the case of this report, the combined data set consists of 11 columns and approximately 17,000 rows. Tableau allowed for simple yet effective visualizations to be created that summarize insights from the data in a logical, easy-to-interpret manner.

C3. Data cleaning

The first step in preparing the data for analysis was to import the add-on data set containing competitor data. To import this into PostgreSQL, a blank table, ‘comp’, was created with each of the column names contained in the original .csv file. Columns that had a match in the original WGU data were encoded with identical data types for matching later on in the code. With the table created, the data set was then imported using PostgreSQL’s built-in import function. Next, the columns from the new data set that had no matching column in the WGU data set were dropped. As the purpose of this analysis was to compare the two providers, having data for one provider that was not present for the other provider was of no use. The last step in preparing the add-on table was to add an additional column for ‘Provider’, with the value ‘Competitor’ for all observations, that would allow for differentiation in the combined data set.

The internal data set from WGU was provided in multiple tables split up by attribute type. These tables were joined to the main ‘Customer’ table through use of primary keys. To work with this data, a new table, ‘wgu’, was created for only the relevant columns from this data set– that is, the columns that had a matching column in the add-on data set. This required the use of two INNER JOIN statements, as the columns needed were in three different tables. Similar to the ‘comp’ table, the ‘wgu’ table also had a ‘Provider’ column added, with the value ‘WGU Telecom’ for all observations.

With both relevant tables now created, the column names needed to be matched up. There were some syntax changes needed (i.e. changing ‘customerid’ in the ‘comp’ table to ‘customer_id’) to accomplish this. Additionally, some of the columns needed their values re-encoded as they were measured slightly differently for each provider. For example, the ‘comp’ table contained a column ‘SeniorCitizen’, while the ‘wgu’ table contained an ‘Age’ column. This required the ‘wgu’ table to be re-encoded using a CASE statement that changed all ‘age’ values greater than or equal to 65 to ‘Yes’, and all values less than 65 to ‘No’. Similar steps were taken to use CASE statements to match up the ‘wgu’ columns of ‘Marital’ and ‘Children’. Additionally, some of the string values required syntax updates for alignment, such as changing ‘Two year’ in the ‘Payment_type’ column of the ‘comp’ table to ‘Two Year’.

The final step before the tables could be combined vertically with a UNION statement was to reorder each table’s columns so that they were in the same order. This was done by simply creating two new tables, ‘wguii’ and ‘compii’, with the columns listed in the appropriate order. With this complete, a new table called ‘combined_data’ was created from a UNION statement between ‘compii’ and ‘wguii’.

With the data now combined, steps were then taken to ensure the final table was clean. First, a COUNT statement was run on all columns to find any missing (IS NULL) values. This statement returned no output, confirming there were no missing values in the combined data set. Next, duplicates of ‘customer_id’ were searched for using a COUNT statement to return the number of duplicates in this column. This statement also returned no output, confirming there were no duplicate values. Finally, the numeric columns in the combined data set, ‘tenure’ and ‘monthly_charge’ were checked for any outliers. The range of values in each column was assessed using the MIN and MAX aggregate functions, respectively. All values for both ‘tenure’ and ‘monthly_charge’ were found to be in acceptable ranges, meaning all values are plausible and do not appear to be the cause of human or machine error. With each of these cleaning steps complete, the final combined data set was ready to be connected to Tableau.

C4. Dashboard creation

The following is a summary of the steps taken to create the dashboards:

1. The data sets are housed in PostgreSQL, where it was combined and cleaned.
2. PostgreSQL was connected to Tableau Desktop Public Edition so the appropriate table could be selected from the database.
3. Multiple worksheets, with various visualizations, were created from the combined data set. These are outlined below:

- *Overall Churn Rates*: ‘Provider’ is used in a quick table calculation to compute the percentage of total churn (CNT(Provider)). Two pie charts, one for each provider, are shown. ‘Provider’ is used as a label mark, and ‘Churn’ is used as a color mark for differentiation. ‘Provider’ is also used in Columns to put the pie charts side by side.

- *Churn Rates by Demo*: This worksheet is constructed the same as the churn % worksheet, with filters for gender, senior citizen, partner, and dependents added and displayed as single-selection filters.
- *HH Tenure*: ‘Provider’ is used in Columns, with ‘Gender’, ‘Has Partner’, ‘Seniorcitizen’, and ‘Has Dependents’ used in Rows. AVG(Tenure) is a calculated field used for color. ‘Square’ is selected from the Marks dropdown to create a heatmap. AVG(Tenure) is also used for the label, to display the values for respective combinations.
- *HH Charges*: ‘Provider’ is used in Columns, with ‘Gender’, ‘Has Partner’, ‘Seniorcitizen’, and ‘Has Dependents’ used in Rows. AVG(Monthly Charge) is a calculated field used for color. ‘Square’ is selected from the Marks dropdown to create a heatmap. AVG(Monthly Charge) is also used for the label, to display the values for respective combinations.
- *Contract Types*: ‘Duration’ and ‘Provider’ are used in Columns, as is. A calculated field, to get the count of provider (CNT(Provider)) is used in Rows. ‘Provider’ is used as the color mark to distinguish them on the bar graph.
- *Tenure by Duration*: ‘Duration’ and ‘Provider’ are used in Columns, as is. A calculated field, to get the average tenure length (AVG(Tenure)) is used in Rows. ‘Provider’ is used as the color mark to distinguish them on the bar graph.
- *Avg Monthly Charge*: AVG(Monthly Charge), a calculated field from ‘Monthly Charge’ is used in Columns, and as label marks to display the average monthly charge for each provider. ‘Provider’ is used in Rows and as a color mark for differentiation.
- *Avg Tenure*: AVG(Tenure), a calculated field from ‘Tenure’ is used in Columns, and as label marks to display the average tenure for each provider. ‘Provider’ is used in Rows and as a color mark for differentiation.
- *Avg Monthly Charge by Churn*: AVG(Monthly Charge), a calculated field from ‘Monthly Charge’ is used in Columns, and as label marks to display the average monthly charge for each provider. ‘Provider’ and ‘Churn’ are used in Rows. ‘Churn is used as a color mark for differentiation.
- *Avg Tenure by Churn*: AVG(Tenure), a calculated field from ‘Tenure’ is used in Columns, and as label marks to display the average tenure for each provider. ‘Provider’ and ‘Churn’ are used in Rows. ‘Churn is used as a color mark for differentiation.

4. The completed worksheets were used to create individual dashboards
   Churn rate + Churn by Demo → *Churn Rates*
   Household tenure + household charges → *Household Member Data*

Contract types + Tenure X Duration → *Contract Data*
Avg Monthly charge + Avg Tenure2 + Avg monthly charge by churn + avg tenure by churn → *Tenure & Charge*

5. The four individual dashboards were added onto a storyboard for ease of use and analysis.

C5. Data analysis results

Results from each dashboard, along with applicable suggestions for action items, are outlined below for use by executive decision-makers at WGU Telecom.

Churn rates:

- Overall churn rates are almost identical for WGU and its competitor. WGU retains 73.5% of its customers, with the competitor retaining 73.46%.
- Notable differences in churn based on demographic data:
    - WGU's competitor has a churn rate almost twice as high as WGU for its senior customers.
    - Our competitor has a higher churn rate for single customers ('No partner').
    - For customers with families (Partner and Dependents both 'Yes'), the competitor has about a 10% lower churn rate than WGU.
        - **Suggestion: WGU Telecom can improve its offerings for families by introducing a wider range of TV or movie streaming content. Additionally, features such as screen-time monitoring or child safety filters could be attractive offerings for families.**

Household member data:

- WGU's average tenure rates are consistent across household member types.
- Average tenure lengths for our competitor's customers who have partners is markedly higher than any other combination of household member type.
    - **Suggestion: WGU Telecom can benefit from recruiting and retaining more customers with partners, as these customers presumably have a tendency to stick with a provider for longer.**.
- WGU's monthly charges are much higher for each household group type than the competitor's. Rates are consistent for customer types among each provider.
    - **Suggestion: If WGU Telecom can lower prices across the board while still remaining profitable, there is potential to reduce churn rate by staying competitive in the telecommunications space.**

Contract data:

- The most popular contract type for both providers is month-to-month.
- One and two year contracts are roughly equally as popular as one another.

- The longest average tenure length for all groups is our competitor's customers with two-year contracts. This is logical, as those with two year contracts are obligated to a longer term with their provider.
    - The next highest average tenure is competitor's customers with one-year contracts. This also makes sense, as customers are locked into this contract duration.
- WGU's average tenure length among its different contract variations is roughly equivalent.
    - **Suggestion: WGU Telecom could benefit from placing more strict rules around breaking contracts. If a customer wishes to switch providers, and they can break a contract with little penalty, then they will likely do so. Therefore, imposing more stringent rules around backing out of contracts could help us increase our average tenure length.**

Tenure & charge:

- WGU's average monthly charge is over twice that of our competitor.
- The highest average monthly charge is associated with WGU customers who churned. This value of $199 is almost $30 more than the average WGU customer.
    - **Suggestion: WGU Telecom could benefit from lowering prices. Reducing our monthly charges could help us be more competitive in the telecommunications market, and thus help with recruitment and retention, both areas we are focused on improving.**
- Average tenure for all WGU customers is slightly higher (roughly 2 months, on average) than our competitor's average tenure.
- There is a greater difference in the range of average tenure lengths for WGU customers than the range for our competitor's customers.

C6. Analysis limitations

The competitor data being analyzed here does not contain as much detail (as many columns, or attributes) as the internal WGU data, limiting the comparisons that could be made. For instance, the WGU data contains information on customer location, income, and job title that the competitor does not provide for their customers. This slightly hinders the making of comparisons of the data sets.

An additional limitation that is potentially present is the lack of information regarding dates of the data collection. It would be most effective to analyze data covering the same time frame for each provider. While there is no information on the dates that the WGU Telecom data set covers, it is noted in accessing the competitor's data set that their data was collected in 2022. For the purposes of this analysis, it was necessary to assume that both data cover the same time frame.

D. Web sources

(n.d.). *SQL UNION Operator*. W3 Schools. Retrieved January 23, 2024, from https://www.w3schools.com/sql/sql_union.asp

E. Sources

Malik, U., Goldwasser, M., & Johnston, B. (2019). *SQL for Data Analytics: Perform Fast and Efficient Data Analysis with the Power of SQL*. Packt Publishing. https://eds.p.ebscohost.com/eds/ebookviewer/ebook?sid=93d41840-5690-47a7-bbe6-d06ae5587727%40redis&ppid=Page-__-1&vid=0&format=EK

Mehfooz, A. (n.d.). *TelecomChurnInsights 2022*. Kaggle. Retrieved January 23, 2024, from https://www.kaggle.com/datasets/ashirzaki/telecomchurninsights

(n.d.). *Why choose Tableau?* Tableau. Retrieved January 24, 2024, from https://www.tableau.com/why-tableau

F. Appendix: SQL code used

```
/*Create blank table for external data*/
CREATE TABLE comp (
CustomerID text,
Gender text,
SeniorCitizen text,
Has_Partner text,
Has_Dependents text,
TenureMonths numeric,
Has_PhoneService varchar(3),
Has_MultipleLines varchar(16),
InternetServiceType varchar(11),
Has_OnlineSecurity varchar(19),
Has_OnlineBackup varchar(19),
Has_DeviceProtection varchar(19),
Has_TechSupport varchar(19),
Has_StreamingTV varchar(19),
Has_StreamingMovies varchar(19),
ContractType text,
PaperlessBilling varchar(3),
PaymentMethodType text,
MonthlyCharges numeric,
Churned text
);

SELECT *
FROM comp
LIMIT 5;

/*Drop columns from comp that WGU data does not have*/
ALTER TABLE comp
DROP COLUMN Has_PhoneService;

ALTER TABLE comp
DROP COLUMN Has_MultipleLines;

ALTER TABLE comp
DROP COLUMN InternetServiceType;

ALTER TABLE comp
DROP COLUMN Has_OnlineSecurity;

ALTER TABLE comp
DROP COLUMN Has_OnlineBackup;

ALTER TABLE comp
DROP COLUMN Has_DeviceProtection;

ALTER TABLE comp
DROP COLUMN Has_TechSupport;

ALTER TABLE comp
DROP COLUMN Has_StreamingTV;

ALTER TABLE comp
DROP COLUMN Has_StreamingMovies;

ALTER TABLE comp
DROP COLUMN PaperlessBilling;

/*Add provider column to differentiate data*/
ALTER TABLE comp
ADD provider varchar(12);

UPDATE comp
SET provider = 'Competitor';
```

```
SELECT *
FROM comp
LIMIT 5;

/* Create table for wgu data through joins using data that aligns
with
external comp data*/
CREATE TABLE wgu AS
SELECT
customer_id,
gender,
age,
marital,
children,
tenure,
duration,
payment_type,
monthly_charge,
churn
FROM customer
INNER JOIN contract
     ON customer.contract_id = contract.contract_id
INNER JOIN payment
     ON customer.payment_id = payment.payment_id;

SELECT *
FROM wgu
LIMIT 5;

/*Match up column names*/
ALTER TABLE comp
RENAME COLUMN customerid TO customer_id;

ALTER TABLE comp
RENAME COLUMN tenuremonths TO tenure;

ALTER TABLE comp
RENAME COLUMN contracttype TO duration;

ALTER TABLE comp
RENAME COLUMN paymentmethodtype TO payment_type;

ALTER TABLE comp
RENAME COLUMN monthlycharges TO monthly_charge;
```

```
ALTER TABLE comp
RENAME COLUMN churned TO churn;

/*Add provider column to differentiate data*/
ALTER TABLE wgu
ADD provider varchar(12);

UPDATE wgu
SET provider = 'WGU Telecom';

SELECT *
FROM wgu
LIMIT 5;

/*Re-encode values for age, marital, children*/
ALTER TABLE wgu
ADD COLUMN seniorcitizen text;

UPDATE wgu
SET seniorcitizen = CASE
     WHEN age >= 65 THEN 'Yes'
     WHEN age < 65 THEN 'No'
     END;

ALTER TABLE wgu
ADD COLUMN has_partner text;

UPDATE wgu
SET has_partner = CASE
     WHEN marital = 'Widowed' OR marital = 'Separated' OR marital =
'Never Married' OR
     marital = 'Divorced' THEN 'No'
     WHEN marital = 'Married' THEN 'Yes'
     END;

ALTER TABLE wgu
ADD COLUMN has_dependents text;

UPDATE wgu
SET has_dependents = CASE
     WHEN children > 0 THEN 'Yes'
     WHEN children = 0 THEN 'No'
     END;

ALTER TABLE wgu
```

```
DROP COLUMN age;

ALTER TABLE wgu
DROP COLUMN marital;

ALTER TABLE wgu
DROP COLUMN children;

SELECT *
FROM wgu
LIMIT 5;

/*Create new wgu table with columns in correct order for performing
UNION*/
CREATE TABLE wguii AS
SELECT customer_id, gender, seniorcitizen, has_partner,
has_dependents,
tenure, duration, payment_type, monthly_charge, churn, provider
FROM wgu;

/*Re-encode comp seniorcitizen values so they match wguii*/
ALTER TABLE comp
ADD COLUMN seniorcitizen_text text;

UPDATE comp
SET seniorcitizen_text = CASE
     WHEN seniorcitizen = '1' THEN 'Yes'
     WHEN seniorcitizen = '0' THEN 'No'
     END;

SELECT *
FROM comp
LIMIT 5;

/*duration and payment_type in comp need to be re-encoded to
match spelling of wguii*/
UPDATE comp
SET duration = CASE
     WHEN duration = 'Two year' THEN 'Two Year'
     WHEN duration = 'One year' THEN 'One year'
     WHEN duration = 'Month-to-month' THEN 'Month-to-month'
     END;

UPDATE comp
SET payment_type = CASE
```

```
     WHEN payment_type = 'Electronic check' THEN 'Electronic Check'
     WHEN payment_type = 'Credit card (automatic)' THEN 'Credit Card
Automatic'
     WHEN payment_type = 'Mailed check' THEN 'Mailed Check'
     WHEN payment_type = 'Bank transfer (automatic)' THEN 'Bank
Transfer Automatic'
     END;

/*Make sure tenure and monthly_charge have same decimal points*/
UPDATE wguii
SET tenure = ROUND(tenure,0);

UPDATE wguii
SET monthly_charge = ROUND(monthly_charge,2)

SELECT *
FROM comp
LIMIT 5;

ALTER TABLE comp
DROP COLUMN seniorcitizen;

ALTER TABLE comp
RENAME COLUMN seniorcitizen_text TO seniorcitizen;

/*Create new wgu table with columns in correct order for performing
UNION*/
CREATE TABLE compii AS
SELECT customer_id, gender, seniorcitizen, has_partner,
has_dependents,
tenure, duration, payment_type, monthly_charge, churn, provider
FROM comp;

/*Joining the two updated tables using UNION and save as new table*/
CREATE TABLE combined_data AS
SELECT * FROM compii
UNION
SELECT * FROM wguii;

SELECT DISTINCT provider
FROM combined_data;

/*Combined data can be cleaned*/
/*Find any missing values*/
SELECT COUNT(*) AS missing_values
```

```sql
FROM combined_data
WHERE customer_id IS NULL OR
gender IS NULL OR
seniorcitizen IS NULL OR
has_partner IS NULL OR
has_dependents IS NULL OR
tenure IS NULL OR
duration IS NULL OR
payment_type IS NULL OR
monthly_charge IS NULL OR
churn IS NULL OR
provider IS NULL;

/*Find any duplicates of customer_id, the unique identifier (primary
key)*/
SELECT customer_id, COUNT(*)
FROM combined_data
GROUP BY customer_id
HAVING COUNT(*) > 1;

/*Find any outliers in the numeric columns: look at range*/
SELECT MAX(tenure)
FROM combined_data;

SELECT MIN(tenure)
FROM combined_data;

SELECT MAX(monthly_charge)
FROM combined_data;

SELECT MIN(monthly_charge)
FROM combined_data;

/*No missing values, duplicates, or outliers were found. None to
treat. Data is clean.*/
```