Web Analytics Report
Bar Dough's Data-Driven Insights

December 20th, 2023

Instructed by Professor Yilu Zhou
Adrian Valentine, Liam Salmon, Sigfús Árnason

I. Executive Summary

Bar Dough is a restaurant located on Restaurant Row in the iconic cuisine district of Hell's Kitchen in New York City. After consulting with the owners of Bar Dough, our team pitched several capabilities of web analytic tools to help the restaurant maintain its competitive edge in the industry. The owners addressed specific challenges faced by the restaurant, including optimizing the menu to boost sales, evaluating the profitability of the happy hour menu, and assessing whether late-night kitchen hours were worth it. While the original proposal simply began as a task for our Web Analytics final project, it has since turned into a significant opportunity for our team to leverage a gap in the restaurant industry. Every restaurant collects sales data; however, few owners have the time or expertise to analyze it effectively. ToastTab is a widely used data management software in the restaurant industry for managing customer transactions. The software lacks user-friendly features for management to analyze this data efficiently. Restaurant owners are restrained from leveraging their data to streamline business operations and improve efficiency.

II. Business Goal Analysis

As business analysts, our role is to simplify the process of interpreting raw sales data and enable restaurant owners to utilize insights to increase sales and decrease expenses. This project focuses on leveraging data to create a better-managed restaurant with a menu tailored to customer preference. The project will utilize web analytics to address the three following challenges:

Menu Optimization: By analyzing customer behaviors, preferences, and trends, our team identified the most profitable and popular menu items at Bar Dough and offered insightful recommendations for items that could benefit from a price change.

Happy Hour Menu Alteration: Leveraging the existing data on Bar Dough's Happy Hour menu to help the restaurant enhance its offering by extending the hours and adding high-margin food items to the list.

Operational Hours Reconstruction: Utilizing the data for the average first purchase and last purchase time in combination with the payroll data, the team offered potential adjustments to the kitchen's operational hours. This strategy aims to reduce staffing costs during periods of low customer activity.

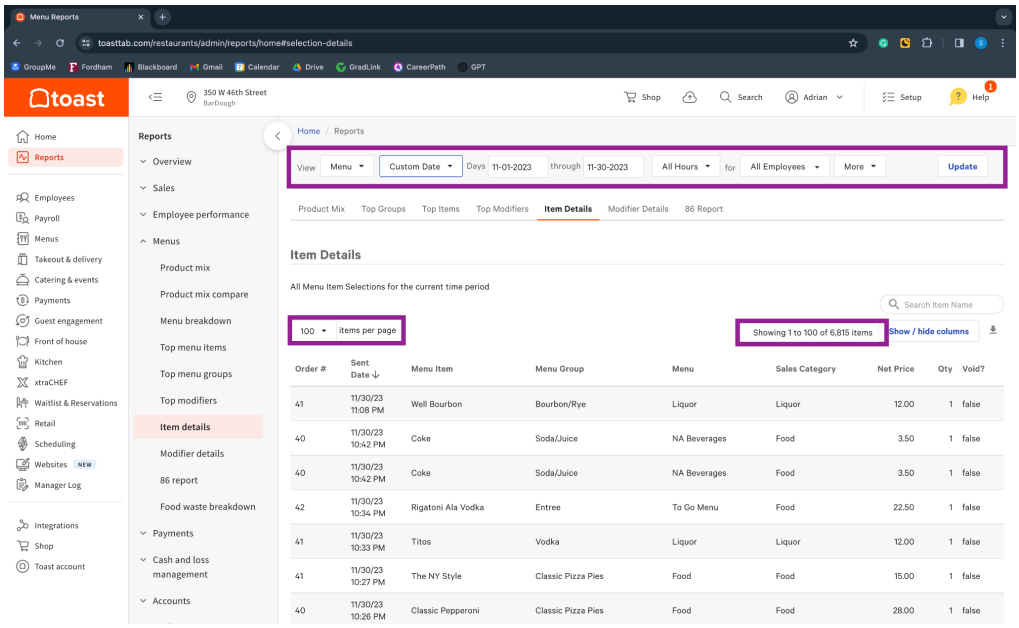
Web analytics is pivotal in the initial data collection phase of this project. Due to the restaurant's outsourced data management software to *Toast Inc*, as discussed further in Section III, the data extraction process required the use of *Selenium* to scrape the website. If the restaurant owners did not have experience with web analytics, it would be challenging to understand Bar Dough's trends because *Toast Inc* has several restrictions on exporting data from its server. Ultimately, our expertise in web scraping was required to complete the data extraction.

III. Dataset Description - Collection and Cleaning:

Once the team had met with the restaurant owners and listened to what they had to say about both the project and potential areas to look into, we had a better understanding of what we wanted and needed from our data collection. The owners created an admin account for us on Bar Dough's ToastTab page. *Toast Inc* is a cloud-based restaurant management software with a plethora of relevant information to offer regarding restaurant management. Upon exploring all the extensive restaurant data, we were able to narrow down what we wanted to extract into two sections, the Item Details page and the Time Entries page.

A. The Item Details page provided us with information relevant to food and drink items on offer at the restaurant, such as the order association, item category, item name, price, quantity ordered, etc.
B. The Time Entries page allowed us to view all employee information, such as role/title, shift times, and wages.

Once we were ready to extract the data, we realized there was a sizable problem with ToastTab as it limited data extraction to 100 results at a time. This became an issue when factoring in the team would have to then download and stitch together between 40 and 85 100-column .csv files for each month to create a year-long data frame.



After speaking with Professor Zhou, she suggested looking into using *Selenium,* which we were able to implement successfully after a few painstaking hours figuring out how to create the code to crawl the data. The process for crawling the data is as follows:

1. The team logged into the ToastTab account
    a. This step involved downloading the required packages for this Selenium crawling, such as Pandas, BeautifulSoup, Selenium, and Time. We were then able to implement the necessary code to log into the webpage as can be seen below.

```
# Open the web page that we want open and log in.
path = "/Users/sigfus/Desktop/Fordham MSBA/Fall 2023/Web Analytics/Project/Selenium/chromedriver"
s = Service(path)
driver = webdriver.Chrome(service = s)
driver.get("https://www.toasttab.com/login")
```

```
# Locate the username input field
username = driver.find_element(By.ID, 'username')
```

```
# Sign in with email first
userid = '[redacted]'
username.send_keys(userid)
```

```
# Click the sign in button to prompt password
sign_in_button = driver.find_element('xpath', '/html/body/div[2]/main/section/div/div/div/div/div/form/div[2]/bu
sign_in_button.click()
```

```
# Locate the password input field
password = driver.find_element(By.ID, 'password')
```

```
# Sign in with password second
key = '[redacted]'
password.send_keys(key)
```

```
# Click the sign in button to enter page
sign_in_button = driver.find_element('xpath', '/html/body/div[2]/main/section/div/div/div/form/div[3]/button')
sign_in_button.click()
```

```
# Get url of Item Detail page (under reports --> menu)
url = 'https://www.toasttab.com/restaurants/admin/reports/home#selection-details'
driver.get(url)
```

2. The team manually ran the automated crawling function
    a. This was by far the most time-intensive part of the project, as code had to be created to extract all 100 search results on a given page before clicking to the next page and repeating the process for each of the pages in each month's item detail and time entries tables.
    b. This was done with a while loop set to run once for each page, and the length of this loop was manually changed for every month based on item count. For each page, the odd and even rows were crawled and extracted into individual and then combined lists. The results of the combined list were then filtered into the various smaller column name lists so that each column in the data table had its own values. These page lists with 100 entries were then appended to the larger monthly lists. After this was done, a timer function was run to skip to the bottom of the page and then click the next page button, where the process was repeated for the duration of the while loop. The time.sleep function proved difficult to gauge as a short time properly often led to duplicate pages, while long times led to very long crawling times.

```
# Crawl item details month by month

nov_order_number = []
nov_sent_date = []
nov_menu_item = []
nov_menu_group = []
nov_menu = []
nov_sales_category = []
nov_net_price = []
nov_quantity = []
nov_void = []

page = 0
while page < 69:

    # Scan page
    soup = bs(driver.page_source)

    # Crawl page
    ## Odd rows
    odd_rows = soup.find_all(class_ = 'odd')
    odd_list = []
    for row in odd_rows:
        # Find all cells in the row and loop through them
        odd_cells = row.find_all('td')
        for cell in odd_cells:
            # Extract text from each cell and convert to int if possible
            odd_text = cell.get_text()
            odd_list.append(odd_text)
    ## Even rows
    even_rows = soup.find_all(class_ = 'even')
    even_list = []
    for row in even_rows:
        # Find all cells in the row and loop through them
        even_cells = row.find_all('td')
        for cell in even_cells:
            # Extract text from each cell and convert to int if possible
            even_text = cell.get_text()
            even_list.append(even_text)
    # Combine lists
    total_list = odd_list + even_list

    # Sort and append lists
    order_number = total_list[::9]
    nov_order_number.append(order_number)

    sent_date = total_list[1::9]
    nov_sent_date.append(sent_date)

    menu_item = total_list[2::9]
    nov_menu_item.append(menu_item)

    menu_group = total_list[3::9]
    nov_menu_group.append(menu_group)

    menu = total_list[4::9]
    nov_menu.append(menu)

    sales_category = total_list[5::9]
    nov_sales_category.append(sales_category)

    net_price = total_list[6::9]
    nov_net_price.append(net_price)

    quantity = total_list[7::9]
    nov_quantity.append(quantity)

    void = total_list[8::9]
    nov_void.append(void)
```

```
# Click to next page
time.sleep(5)
#Scroll down to the bottom of the page
driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
#wait for page to be loaded
time.sleep(5)
#find and click 'next' button
next_page_button = driver.find_element('xpath', '//*[@id="menu-items-report_wrapper"]/div[4]/div/div/ul/li[7]/a
next_page_button.click()
time.sleep(5)

page += 1
```

3. The team then checked for duplicate values
    a. Once all the data was collected for a given month, a nested loop function was used to check for duplicate lists. The larger monthly lists still consisted of 100 entry smaller lists within the larger list (nested lists). This nested loop function allowed me to compare if any of these nested lists were equal, as this would have indicated an error in the crawling process. This happened a few times and was often a result of the timer being set to too fast a speed and/or a poor internet connection, resulting in too little time for the function to crawl a given page before clicking next to the following page.

```
# Check for duplicate pages (error in crawling/page switching)
# For loop to select page's list
for i in range(len(nov_order_number)):
    # Nested for loop to go through other pages' lists
    for j in range(i+1, len(nov_order_number)):
        # Evaluation to see if page i's data is equal to other lists' data
        if nov_order_number[i] == nov_order_number[j]:
            # Show where the duplicates are found
            print("The lists at index {} and {} are equal: {}".format(i, j)
```

4. The team then expanded the nested lists
    a. This was a straightforward step used to take the nested lists and in a sense, un-nesting them so that each monthly list no longer had a nested list for each column on each page but rather a continuous singular list of values for each column incorporating all the pages.

```
# Combine page lists into one huge list
complete_nov_order_number = [inner_item for outer_item in nov_order_number for inner_item in outer_item]
complete_nov_sent_date = [inner_item for outer_item in nov_sent_date for inner_item in outer_item]
complete_nov_menu_item = [inner_item for outer_item in nov_menu_item for inner_item in outer_item]
complete_nov_menu_group = [inner_item for outer_item in nov_menu_group for inner_item in outer_item]
complete_nov_menu = [inner_item for outer_item in nov_menu for inner_item in outer_item]
complete_nov_sales_category = [inner_item for outer_item in nov_sales_category for inner_item in outer_item]
complete_nov_net_price = [inner_item for outer_item in nov_net_price for inner_item in outer_item]
complete_nov_quantity = [inner_item for outer_item in nov_quantity for inner_item in outer_item]
complete_nov_void = [inner_item for outer_item in nov_void for inner_item in outer_item]
```

5. The team then created the final data frames
    a. Once we had the monthly lists, we were able to combine them into pandas data frames, first for each given month and then for the whole year.

```
# Create data frame for month
nov_item_details = {'Order Number': complete_nov_order_number,
                    'Sent Date': complete_nov_sent_date,
                    'Menu Item': complete_nov_menu_item,
                    'Menu Group': complete_nov_menu_group,
                    'Menu': complete_nov_menu,
                    'Sales Category': complete_nov_sales_category,
                    'Net Price': complete_nov_net_price,
                    'Quantity': complete_nov_quantity,
                    'Void?': complete_nov_void
                    }

nov_item_details = pd.DataFrame(nov_item_details)
nov_item_details.sort_values(by = "Sent Date")
nov_item_details.shape
```

```
(6815, 9)
```

```
# Create a list of the 12 monthly data frames
item_details_seperate = [dec_22_item_details, jan_item_details, feb_item_details,
                         mar_item_details, apr_item_details, may_item_details,
                         jun_item_details, jul_item_details, aug_item_details,
                         sep_item_details, oct_item_details, nov_item_details]
```

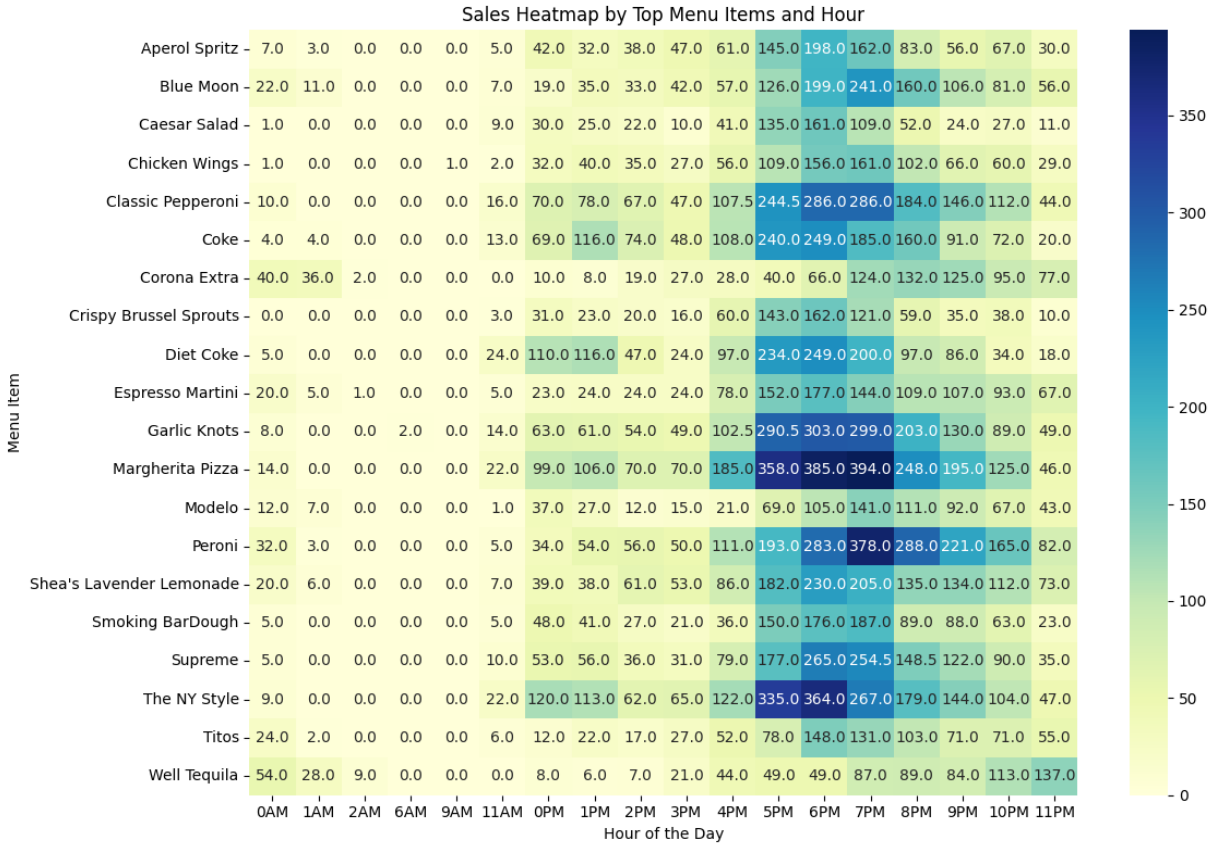
```
# Combine them into one data frame with a new index rather than index from monthly data frames
item_details_full = pd.concat(item_details_seperate, ignore_index=True)
item_details_full.shape
```

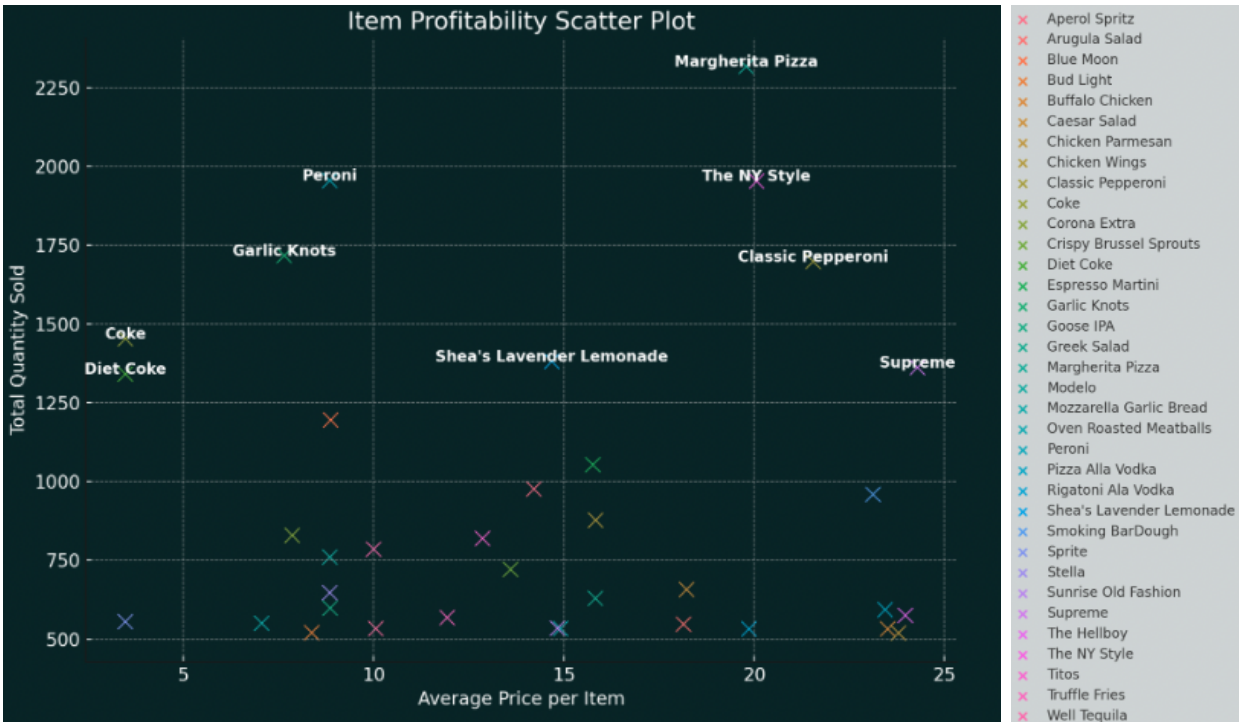
```
(67459, 9)
```

With this process complete, the team now had two extensive data sets for item details and time entries containing around 68,000 and 2,300 entries, respectively. We were then able to clean the data by having the right type associated with each column and splitting up the “Date” column in item details by creating two columns for Date and Time as opposed to having them both in the same column.

IV. Menu Optimization

In our analysis of Menu Optimization for Bar Dough, the team looked at customer behaviors, preferences, and the latest trends to formulate recommendations. The team identified several key menu items that are not only popular among customers but also contribute significantly to the restaurant’s profitability. This examination provides Bar Dough with the data needed to make well-informed recommendations. Specifically, the team has selected certain menu items that could benefit from a price adjustment. The following two charts illustrate the sales heatmap for total items sold by hour. Each metric represents the total number of items sold by hour in the year’s worth of data.


Sales Heatmap by Top Menu Items and Hour
Menu Item
Hour of the Day
0AM 1AM 2AM 6AM 9AM 11AM 0PM 1PM 2PM 3PM 4PM 5PM 6PM 7PM 8PM 9PM 10PM 11PM
Aperol Spritz 7.0 3.0 0.0 0.0 0.0 5.0 42.0 32.0 38.0 47.0 61.0 145.0 198.0 162.0 83.0 56.0 67.0 30.0
Blue Moon 22.0 11.0 0.0 0.0 0.0 7.0 19.0 35.0 33.0 42.0 57.0 126.0 199.0 241.0 160.0 106.0 81.0 56.0
Caesar Salad 1.0 0.0 0.0 0.0 0.0 9.0 30.0 25.0 22.0 10.0 41.0 135.0 161.0 109.0 52.0 24.0 27.0 11.0
Chicken Wings 1.0 0.0 0.0 0.0 1.0 2.0 32.0 40.0 35.0 27.0 56.0 109.0 156.0 161.0 102.0 66.0 60.0 29.0
Classic Pepperoni 10.0 0.0 0.0 0.0 0.0 16.0 70.0 78.0 67.0 47.0 107.5 244.5 286.0 286.0 184.0 146.0 112.0 44.0
Coke 4.0 4.0 0.0 0.0 0.0 13.0 69.0 116.0 74.0 48.0 108.0 240.0 249.0 185.0 160.0 91.0 72.0 20.0
Corona Extra 40.0 36.0 2.0 0.0 0.0 0.0 10.0 8.0 19.0 27.0 28.0 40.0 66.0 124.0 132.0 125.0 95.0 77.0
Crispy Brussel Sprouts 0.0 0.0 0.0 0.0 0.0 3.0 31.0 23.0 20.0 16.0 60.0 143.0 162.0 121.0 59.0 35.0 38.0 10.0
Diet Coke 5.0 0.0 0.0 0.0 0.0 24.0 110.0 116.0 47.0 24.0 97.0 234.0 249.0 200.0 97.0 86.0 34.0 18.0
Espresso Martini 20.0 5.0 1.0 0.0 0.0 5.0 23.0 24.0 24.0 24.0 78.0 152.0 177.0 144.0 109.0 107.0 93.0 67.0
Garlic Knots 8.0 0.0 0.0 2.0 0.0 14.0 63.0 61.0 54.0 49.0 102.5 290.5 303.0 299.0 203.0 130.0 89.0 49.0
Margherita Pizza 14.0 0.0 0.0 0.0 0.0 22.0 99.0 106.0 70.0 70.0 185.0 358.0 385.0 394.0 248.0 195.0 125.0 46.0
Modelo 12.0 7.0 0.0 0.0 0.0 1.0 37.0 27.0 12.0 15.0 21.0 69.0 105.0 141.0 111.0 92.0 67.0 43.0
Peroni 32.0 3.0 0.0 0.0 0.0 5.0 34.0 54.0 56.0 50.0 111.0 193.0 283.0 378.0 288.0 221.0 165.0 82.0
Shea's Lavender Lemonade 20.0 6.0 0.0 0.0 0.0 7.0 39.0 38.0 61.0 53.0 86.0 182.0 230.0 205.0 135.0 134.0 112.0 73.0
Smoking BarDough 5.0 0.0 0.0 0.0 0.0 5.0 48.0 41.0 27.0 21.0 36.0 150.0 176.0 187.0 89.0 88.0 63.0 23.0
Supreme 5.0 0.0 0.0 0.0 0.0 10.0 53.0 56.0 36.0 31.0 79.0 177.0 265.0 254.5 148.5 122.0 90.0 35.0
The NY Style 9.0 0.0 0.0 0.0 0.0 22.0 120.0 113.0 62.0 65.0 122.0 335.0 364.0 267.0 179.0 144.0 104.0 47.0
Titos 24.0 2.0 0.0 0.0 0.0 6.0 12.0 22.0 17.0 27.0 52.0 78.0 148.0 131.0 103.0 71.0 71.0 55.0
Well Tequila 54.0 28.0 9.0 0.0 0.0 0.0 8.0 6.0 7.0 21.0 44.0 49.0 49.0 87.0 89.0 84.0 113.0 137.0
0 50 100 150 200 250 300 350

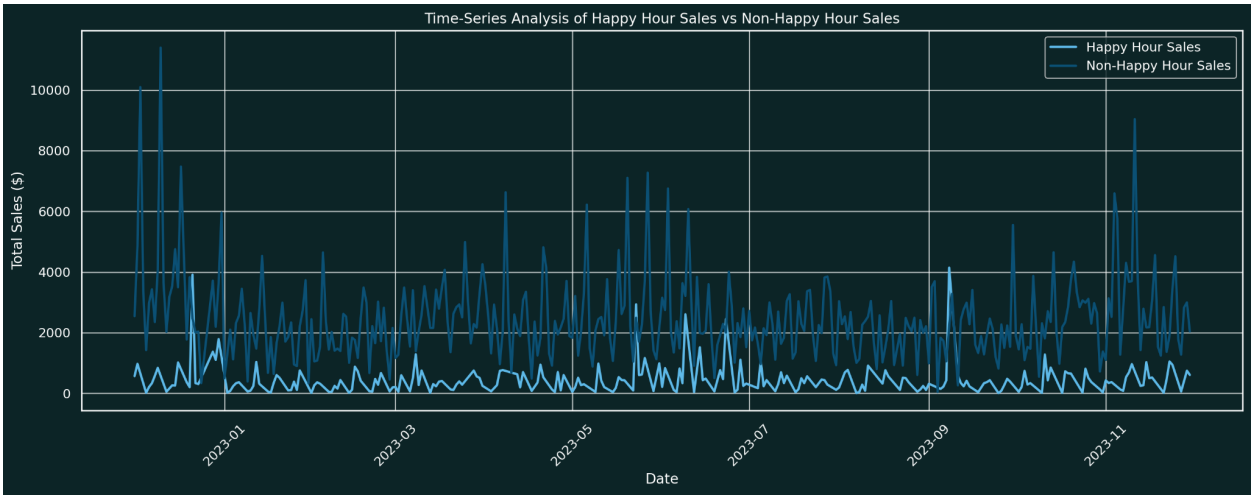

Item Profitability Scatter Plot
Total Quantity Sold
Average Price per Item
Margherita Pizza
Peroni
The NY Style
Garlic Knots
Classic Pepperoni
Coke
Diet Coke
Shea's Lavender Lemonade
Supreme
500 750 1000 1250 1500 1750 2000 2250
5 10 15 20 25
Aperol Spritz
Arugula Salad
Blue Moon
Bud Light
Buffalo Chicken
Caesar Salad
Chicken Parmesan
Chicken Wings
Classic Pepperoni
Coke
Corona Extra
Crispy Brussel Sprouts
Diet Coke
Espresso Martini
Garlic Knots
Goose IPA
Greek Salad
Margherita Pizza
Modelo
Mozzarella Garlic Bread
Oven Roasted Meatballs
Peroni
Pizza Alla Vodka
Rigatoni Ala Vodka
Shea's Lavender Lemonade
Smoking BarDough
Sprite
Stella
Sunrise Old Fashion
Supreme
The Hellboy
The NY Style
Titos
Truffle Fries
Well Tequila

As a result of the sales heatmap and item profitability scatter plot, the team recommends the following menu changes:

- Increase the price of high-demand items that consistently sell across different time periods to capitalize on popularity and improve profit margins.
    - These items include the Margherita Pizza, The NY Style, and a Classic Pepperoni pizza.
- Include food items in the Happy Hour menu that have lower sales during early hours to increase demand and introduce them to more customers.
    - Bar Dough should reduce the price of Garlic Knots, Crispy Brussel Sprouts, and The Supreme during the hours of 12 pm to 5 pm to attract more customers.
- Bar Dough should consider a limited-time offer for items that have periodic popularity to boost sales during off-peak hours.
    - Bar Dough should introduce a “Midweek Mixer” special where they offer Shea's Lavender Lemonade. The special features a discounted price on the Lemonade during midweek afternoons because of its slow historical trend.

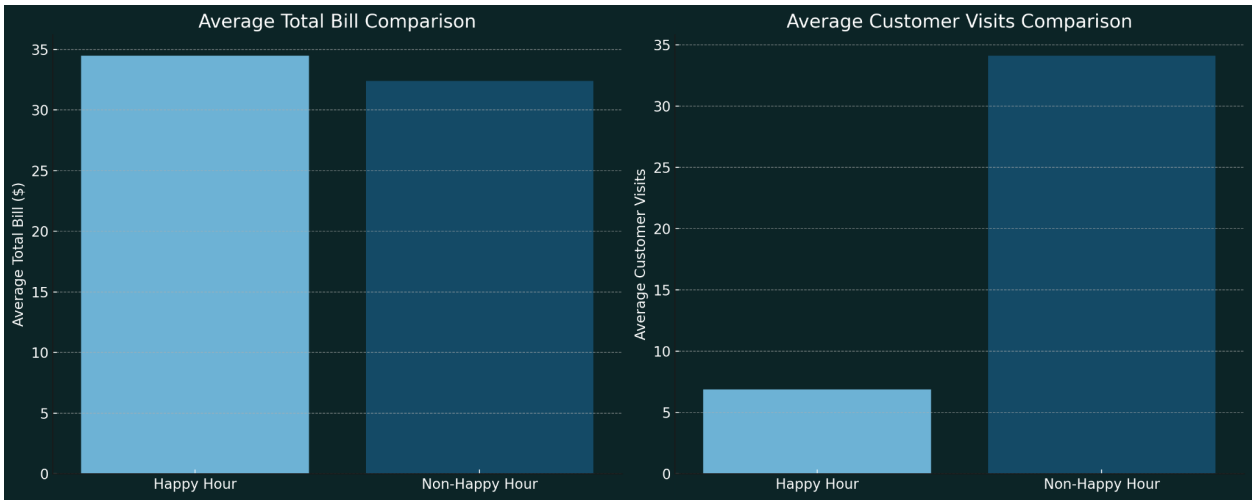
V. Happy Hour Menu Optimization

Sales trends reveal a pattern of fluctuation in Happy Hour versus Non-Happy Hour sales, which suggests customer visits and spending differ during these periods. This inconsistency illustrates the potential for optimizing the scheduling and marketing of Happy Hour to increase foot traffic and items customers want to buy. The chart below shows a time-series analysis of Happy Hour versus Non-Happy Hour sales.

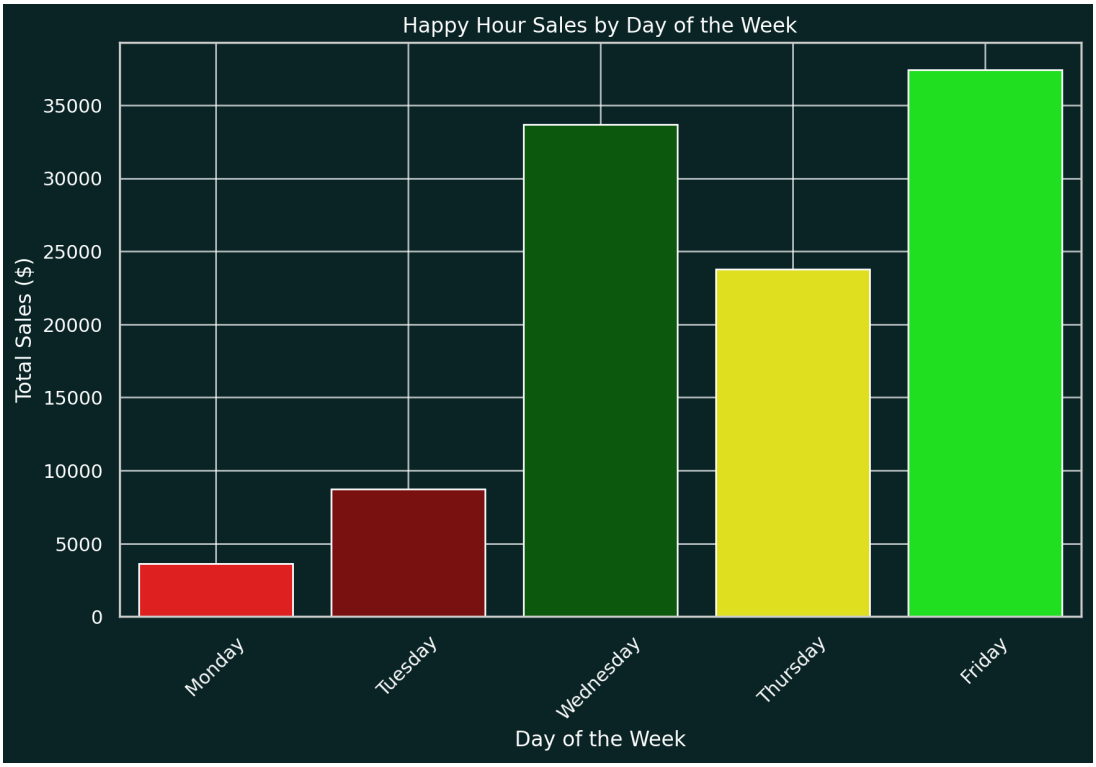


When examining the average bill and customer visits between these two distinct selling periods, performance varies significantly, revealing some interesting takeaways. The average total bill during Happy Hour is $34.49, which exceeds Non-Happy Hour periods, where the average bill is $32.40. As a bartender in the restaurant industry for nearly two years, this is a great sign for Bar Dough. Restaurants, in general, want customers to come in during Happy Hour

for discounted drink prices because they spend more money. The fact Bar Dough has already accomplished this goal is a positive sign. However, the average number of customer visits during Happy Hour is significantly lower, with 6.90 visits per day compared to the 34.14 visits during other times. This discrepancy between the average total bill price and average customer visits suggests that while Happy Hour may attract higher spending, it does not necessarily translate to increased foot traffic.



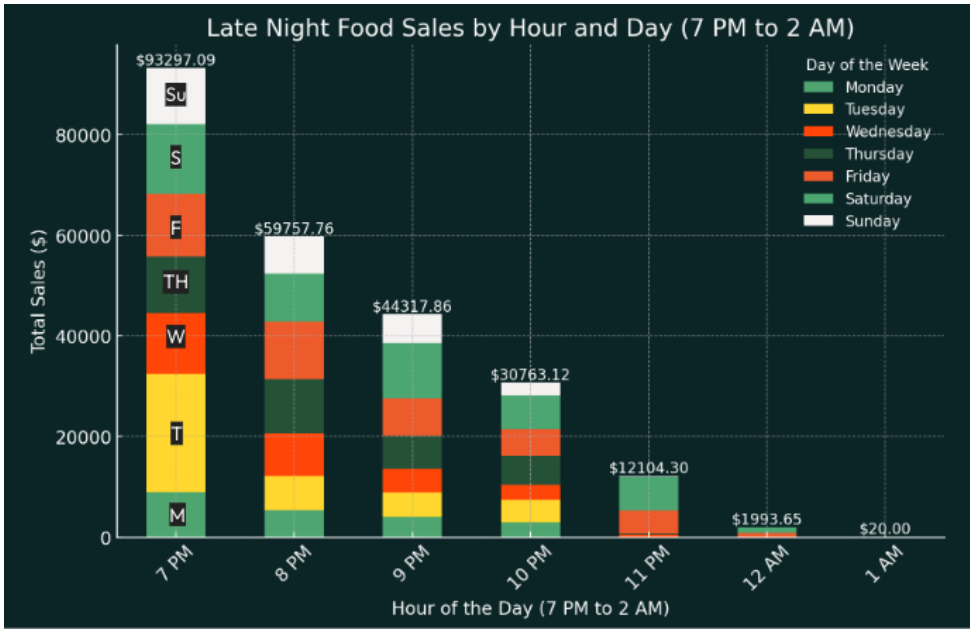
An analysis of Happy Hour sales by the day of the week demonstrates the variability in foot traffic, with Wednesday and Friday emerging as the most lucrative days. This trend may reflect a pattern of customer preference or availability, indicating that these days are a success for attracting customers for Happy Hour. The chart below illustrates how Bar Dough can optimize sales opportunities on specific days of the week–particularly Monday, Tuesday, and Thursday.



Based on these insights, several recommendations are proposed to enhance the effectiveness of Happy Hour. Increasing the visibility of Happy Hour could potentially increase sales on the slower days of Monday, Tuesday, and Thursday. In addition, due to the late kitchen opening on Monday and Tuesday, we suggest reviewing Happy Hour hours on these days to enhance the appeal of these offerings, potentially leading to an increase in the average bill and more foot traffic. Perhaps even offering special Happy Hour events, particularly themed around local games, shows, or concerts. Bar Dough's proximity to Madison Square Garden and Broadway makes it an ideal location for customers of nearby events to come and visit. With an extended Happy Hour, customers could easily be persuaded to visit during these less busy nights.

VI. Operational Hours Reconstruction

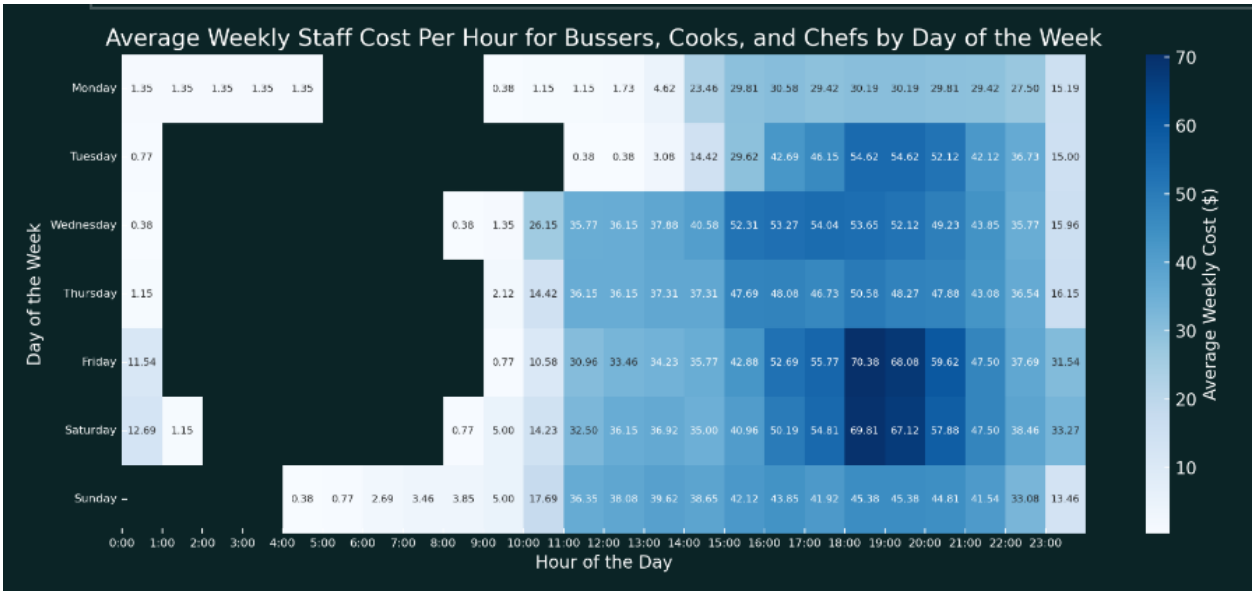
After discussing the profitability of late-night food sales with the Bar Dough owners, we conducted an extensive examination of sales data to determine the most cost-effective and customer-centric operational hours. The following chart shows food trends from the dinner rush until late night, from 7 pm until 2 am. While sales exhibit a relatively consistent distribution at 7 pm throughout the week, there is a notable shift beyond 11 pm. It is apparent that extending kitchen operations beyond 11 pm is justified on Fridays and Saturdays because of the increased customer demand during peak late-night weekend hours.



By analyzing the average first and last purchase times for each day of the week, the team strategically adjusted the kitchen menu hours to align with peak customer activity. The following chart illustrates our data-driven restructuring to optimize the workflow of the kitchen staff and the estimated cost savings. The adjustment not only enhances operational efficiency but also ensures that Bar Dough is well-positioned to provide an enhanced dining experience during high-traffic hours, which fosters customer satisfaction.

Day of the Week	Current Hours	Recommended	Money Saved*
Monday	4:00PM-11:00PM	4:30PM-11:00PM	$15.29
Tuesday	4:00PM-11:00PM	4:30PM-11:00PM	$21.34
Wednesday	11:30AM-11:00PM	12:00PM-11:00PM	$17.88
Thursday	11:30AM-11:00PM	12:30PM-11:00PM	$35.96
Friday	11:30AM-11:30PM	12:30PM-12:00AM	$16.44
Saturday	11:30AM-11:30PM	12:00PM-12:00AM	$0
Sunday	11:30AM-11:00PM	12:00PM-11:00PM	$18.18

Leveraging the average weekly staff cost per hour for the kitchen staff, we calculated that the revised operational hours would save approximately $125.1 per week and an annual total of $6,504.42. While these additional savings might not seem like much for a restaurant in New York City, they hold significant value for Bar Dough as a small-scale restaurant. This could provide a meaningful budget for Happy Hour promotions, particularly during the low-sales periods on Monday and Tuesday. Reconstructing the operational hours of the kitchen will allow Bar Dough to benefit financially and better allocate its resources.



VII. Conclusion

The team's analysis and strategic recommendations for Bar Dough leverage several of the web analytic tools learned throughout the semester. By utilizing *Selenium* to scrape Bar Dough's year-long sales data, the team was able to integrate valuable recommendations to assist the restaurant moving forward. Through a revised menu based on customer preferences, Bar Dough can maximize profitability and customer satisfaction. The team's insights into Happy Hour sales patterns offer actionable strategies to increase customer foot traffic while maintaining the high bill prices. The restructuring of operational hours, based on peak sales hours, promotes cost savings and efficiency for the restaurant. This report highlights the fact that restaurants are moving towards a data-driven industry, where business analytics is replacing years of experience and becoming the focal point of decision-making.

VIII. Future Steps

Bar Dough represents one of eleven restaurants owned by the Allied Group here in New York City. What began as a straightforward academic project has clearly shown potential benefits that could extend to the group's ten other restaurants. Our team aims to leverage the upcoming meeting on January 20th, 2024, into a business proposal where valuable business insights can be discovered and applied to a corporation with an estimated $50 million in revenue per year.

In terms of specific future steps for Bar Dough, it would be interesting to apply additional web analytic tools to understand customer behavior online via Yelp reviews or by tracking the effectiveness of delivery services such as DoorDash, Uber Eats, and Grub Hub. The team could evaluate the current marketing strategy and whether the implementation of a digital marketing campaign tailored to the online customer demographic would be beneficial.