IST2334 - Web and Network Analytics - Practical #4

The following case study includes website performance data from Google. These data summarize website traffic from the first twenty-three weeks of operation of the ToutBay website (April 12 through September 19, 2014). After reading the case study, you are required to analyze the data to provide insights about the browser and operating system usage by website visitors of ToutBay. Each of the analysis steps needs to be saved in a pdf format.

CASE Study: ToutBay Begins

In October 2014, ToutBay LLC, a publisher and distributor of data science applications, remains in start-up mode awaiting the release of its first products. In what is becoming an increasingly data-driven world, ToutBay cofounders Greg Blence and Tom Miller see opportunities for *data science as a service (DSaaS)*, a term they use to describe ToutBay's business. ToutBay's goal is to be a market maker in the data science space, publishing and distributing time-sensitive information and competitive intelligence.

There are four application areas: sports, finance, marketing, and general information. Sports touts go beyond raw data about players and teams to build models that predict future performance. ToutBay works with sports touts to make their predictive models available to players, owners, managers, and sports enthusiasts.

Finance touts help individuals and firms make informed decisions about when and where to make investments. These touts have expertise in econometrics and time series analysis. They understand markets and predictive models. They detect trends in the past and make forecasts about the future.

One of ToutBay's first financial products, the Stock Portfolio Constructor, is the work of Dr. Ernest P. Chan, a recognized expert in the area of quantitative finance and author of two books on the subject (Chan 2009, 2013). The idea behind this product is to allow a stock investor to specify his/her investment objectives and time horizon, as well as the domain of stocks being considered and the number of stocks desired in a portfolio. Then, using current information about stock prices and performance, as well as selected economic factors, the Stock Portfolio Constructor creates a customized stock portfolio for the investor. It lists the selected stocks and shows their expected

future return over the investor's time horizon, assuming an equal level of investment in each stock. The Stock Portfolio Constructor also shows what would have been the historical performance of that portfolio in recent years.

Marketing touts play a similar expert role, going beyond raw sales data to provide consumer and marketplace insights. They have formal training in measurement, statistics, or machine learning, as well as extensive business consulting experience. The results of their models for site selection, product positioning, segmentation, or target marketing are of special interest to business managers.

General science, a fourth product area, involves scientists of all stripes, building models of human interest. ToutBay intends to give scientific thinking and models wider distribution than they are likely to receive when published in academic journals.

ToutBay's major public event to date has been the R User Conference, also known as UseR!, June 30 through July 3, 2014. The conference was held on the UCLA campus in Los Angeles, California, and attracted around 700 scientists and software engineers, people who write programs (scripts) in the open-source language R (a widely used language in statistics and data science). ToutBay was one of the sponsors of UseR!, along with major software developers and publishers.

ToutBay's goal at UseR! was to introduce itself to potential touts. The company's message was simple: *You do the research and modeling, and we do the rest. We turn scripts into products.* The idea is that, by working with ToutBay, data scientists can focus on data science and ToutBay will take care of marketing, communications, sales, order processing, distribution, and customer support. The ToutBay website has a *For Touts* page that provides the details.

Because ToutBay operates entirely online, its business depends on having a website that conveys a clear message to visitors or guests. Success means converting website guests into ToutBay account holders. And after information products become available, success will mean converting account holders into subscribers to information products.

Revenues will come from customer subscriptions, with touts setting prices for their information products and ToutBay charging a fee for online sales and distribution of those products. In recruiting future touts, ToutBay has a simple message: *If you were the author of a book, you would look for a publisher, and you would hope that the publisher would work with bookstores to sell your book. But what if you are the author of a predictive model? Where do you go to publish your model? Where do you go to sell the results of your model? ToutBay—that's where.*

Since opening its website in April 2014, ToutBay has been tracking user traffic with Google Analytics. Recently, the firm has been reviewing data relating to visits, page views, and time on the site. There may have been a slight increase in traffic around the time of the UseR! conference. Otherwise, traffic has been limited, which is a source of concern for the company.

The ToutBay website employs a single-page design, with extensive information on the home page, including the two-minute video introduction to the company. A single-page approach to website design provides better overall performance than a multi-page approach because a single-page approach requires fewer data transmissions between the client browser and the website server.

One difficulty in employing a single-page approach, however, is that standard page-view statistics provide an incomplete picture of website usage. Recognizing this, ToutBay website developers employed JavaScript code to detect how far down users were scrolling on the home page. These scrolling data are included in user traffic information for the site.

Steps for Browser and Operating System Analysis:

1. **Data Preparation**:
    - Import the dataset into a Python environment for analysis.
    - Examine the data structure to identify columns related to browser and operating system usage.
2. **Insights from Browser Usage**:
    - **Top Browsers**: Identify the most popular browsers used by the website visitors during the observed period.
    - **Browser Trends**: Analyze the trends over the weeks to see if browser preferences changed (e.g., more users moving towards Chrome or Firefox).
    - **User Behavior by Browser**: Look for patterns such as the average session duration or bounce rate for different browsers to understand user engagement.

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv('toutbay_website_traffic.csv')

# Display the first few rows of the dataset to confirm the data
print(df.head())

### 1. Top Browsers: Find which browsers have the most sessions
top_browsers = df.groupby('Browser')['Sessions'].sum().reset_index().sort_values(by='Sessions', ascending=False)

# Plot the Top Browsers by Sessions
plt.figure(figsize=(10, 5))
plt.bar(top_browsers['Browser'], top_browsers['Sessions'], color='skyblue')
plt.title('Top Browsers by Sessions')
plt.xlabel('Browser')
plt.ylabel('Number of Sessions')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
# 2. Browser Trends: Analyze browser trends over time
# Convert Date column to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Group by Browser and Date to observe trends over time
browser_trends = df.groupby([df['Date'].dt.to_period('M'), 'Browser'])['Sessions'].sum().unstack()

# Plot Browser Trends
browser_trends.plot(kind='line', figsize=(12, 6))
plt.title('Browser Trends Over Time (Monthly Sessions)')
plt.xlabel('Date')
plt.ylabel('Number of Sessions')
plt.legend(title='Browser')
plt.tight_layout()
plt.show()
```

```
# 3. User Behavior by Browser: Analyzing behavior using bounce rate and session duration
browser_behavior = df.groupby('Browser').agg({
    'Sessions': 'sum',
    'Page Views': 'sum',
    'Bounce Rate': 'mean',
    'Avg Session Duration (sec)': 'mean'
}).reset_index()

# Display browser behavior insights
print(browser_behavior)

# Plotting average bounce rate by browser
plt.figure(figsize=(10, 5))
plt.bar(browser_behavior['Browser'], browser_behavior['Bounce Rate'], color='lightgreen')
plt.title('Average Bounce Rate by Browser')
plt.xlabel('Browser')
plt.ylabel('Bounce Rate')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
# Plotting average session duration by browser
plt.figure(figsize=(10, 5))
plt.bar(browser_behavior['Browser'], browser_behavior['Avg Session Duration (sec)'], color='orange')
plt.title('Average Session Duration by Browser')
plt.xlabel('Browser')
plt.ylabel('Avg Session Duration (seconds)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Key Insights Generated from the above analysis

a. Top Browsers:

- This section ranks the browsers by the number of sessions they generated. The plot will show the top browsers that brought the most traffic to the website.

b. Browser Trends:

- This tracks the usage of different browsers over time, showing how browser preferences changed from April to September.

c. User Behavior by Browser:

- This part analyzes user behavior based on:
    - **Bounce Rate**: The average rate at which users from each browser leave the site after viewing only one page.
    - **Average Session Duration**: How long users from each browser stay on the website on average.

3. **Insights from Operating System Usage**:

- **Top Operating Systems**: Similar to browsers, determine which operating systems (e.g., Windows, macOS, Linux) are most popular among users.
- **Operating System Trends**: Track how operating system usage evolved over the months.
- **OS-Specific Engagement**: Compare metrics like scroll depth, session length, and page views for each operating system.

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv('toutbay_website_traffic.csv')

# Display the first few rows of the dataset to confirm the data
print(df.head())

# 1. Top Operating Systems: Find which operating systems have the most sessions
top_os = df.groupby('Operating System')['Sessions'].sum().reset_index().sort_values(by='Sessions', ascending=False)

# Plot the Top Operating Systems by Sessions
plt.figure(figsize=(10, 5))
plt.bar(top_os['Operating System'], top_os['Sessions'], color='skyblue')
plt.title('Top Operating Systems by Sessions')
plt.xlabel('Operating System')
plt.ylabel('Number of Sessions')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```python
# 2. Operating System Trends: Analyze OS trends over time
# Convert Date column to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Group by OS and Date to observe trends over time
os_trends = df.groupby([df['Date'].dt.to_period('M'), 'Operating System'])['Sessions'].sum().unstack()

# Plot OS Trends
os_trends.plot(kind='line', figsize=(12, 6))
plt.title('Operating System Trends Over Time (Monthly Sessions)')
plt.xlabel('Date')
plt.ylabel('Number of Sessions')
plt.legend(title='Operating System')
plt.tight_layout()
plt.show()
```

```python
# 3. OS-Specific Engagement: Analyze behavior using bounce rate and session duration by OS
os_behavior = df.groupby('Operating System').agg({
    'Sessions': 'sum',
    'Page Views': 'sum',
    'Bounce Rate': 'mean',
    'Avg Session Duration (sec)': 'mean'
}).reset_index()

# Display OS-specific engagement insights
print(os_behavior)

# Plotting average bounce rate by OS
plt.figure(figsize=(10, 5))
plt.bar(os_behavior['Operating System'], os_behavior['Bounce Rate'], color='lightgreen')
plt.title('Average Bounce Rate by Operating System')
plt.xlabel('Operating System')
plt.ylabel('Bounce Rate')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```python
# Plotting average session duration by OS
plt.figure(figsize=(10, 5))
plt.bar(os_behavior['Operating System'], os_behavior['Avg Session Duration (sec)'], color='orange')
plt.title('Average Session Duration by Operating System')
plt.xlabel('Operating System')
plt.ylabel('Avg Session Duration (seconds)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Key Insights generated from the above analysis:

a. Top Operating Systems:

- This ranks the operating systems by the number of sessions generated. The plot will show which OS (e.g., Windows, macOS, Linux) contributed the most traffic to the website.

b. Operating System Trends:

- This shows how the usage of different operating systems evolved over time (from April 12 to September 19, 2014). You can observe how the number of sessions varied by month for each OS.

c. OS-Specific Engagement:

- This part provides insights into the user behavior based on the operating system, including:
    - **Bounce Rate**: The average rate at which users from different operating systems leave the site after viewing just one page.

- **Average Session Duration**: How long users from different operating systems stay on the site.

4. **Combining Browser and OS Data**:

- **Cross-Analysis**: Explore browser and operating system combinations. For example, which browser and OS combination led to the highest conversion rates or longest session durations?

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv('toutbay_website_traffic.csv')

# Display the first few rows of the dataset to confirm the data
print(df.head())

# 1. Cross-Analysis: Sessions by Browser and OS Combination
browser_os_sessions = df.groupby(['Browser', 'Operating System'])['Sessions'].sum().unstack()

# Plot the cross-analysis of sessions by Browser and Operating System
plt.figure(figsize=(12, 6))
sns.heatmap(browser_os_sessions, annot=True, cmap='Blues', fmt="g")
plt.title('Cross-Analysis: Sessions by Browser and Operating System')
plt.xlabel('Operating System')
plt.ylabel('Browser')
plt.tight_layout()
plt.show()
```

```
# 2. Cross-Analysis: Bounce Rate by Browser and OS Combination
browser_os_bounce_rate = df.groupby(['Browser', 'Operating System'])['Bounce Rate'].mean().unstack()

# Plot the cross-analysis of bounce rate by Browser and Operating System
plt.figure(figsize=(12, 6))
sns.heatmap(browser_os_bounce_rate, annot=True, cmap='Reds', fmt=".2f")
plt.title('Cross-Analysis: Bounce Rate by Browser and Operating System')
plt.xlabel('Operating System')
plt.ylabel('Browser')
plt.tight_layout()
plt.show()

# 3. Cross-Analysis: Average Session Duration by Browser and OS Combination
browser_os_session_duration = df.groupby(['Browser', 'Operating System'])['Avg Session Duration (sec)'].mean().unstack()

# Plot the cross-analysis of average session duration by Browser and Operating System
plt.figure(figsize=(12, 6))
sns.heatmap(browser_os_session_duration, annot=True, cmap='Greens', fmt=".0f")
plt.title('Cross-Analysis: Avg Session Duration by Browser and Operating System')
plt.xlabel('Operating System')
plt.ylabel('Browser')
plt.tight_layout()
plt.show()
```

Key Insights Generated from Cross-Analysis

a. Sessions by Browser and OS Combination:

- This heatmap shows the total number of sessions for each combination of browser and operating system (e.g., Chrome on Windows vs. Firefox on macOS).

b. Bounce Rate by Browser and OS Combination:

- This heatmap shows the average bounce rate for each combination of browser and operating system. You can quickly identify which browser-OS combinations lead to higher or lower bounce rates.

c. Average Session Duration by Browser and OS Combination:

- This heatmap shows the average session duration for each browser-OS combination, highlighting which combinations have users spending more time on the website.