CxC 2025

TouchBistro Challenge

By Shami-uz Zaman, Tony Ngo, Roah Cho, Angad Ahluwalia

Table of Contents

1. Introduction

The TouchBistro x UW problem set offers a comprehensive dataset capturing real-world restaurant transactions, including details on venues, bills, orders, tips, and operational timing. This project aims to leverage this dataset to extract actionable insights into operational efficiency, customer behavior, and external factors influencing restaurant performance. Specific analyses include peak period identification for staffing, revenue forecasting, customer spending behavior, and the integration of external variables like weather and inflation data.

Objectives:

- **Operational Efficiency:** Identify peak periods to optimize staffing and improve service delivery.
- **Revenue Forecasting:** Develop models to predict future sales, aiding inventory and staffing decisions.
- **Customer Insights:** Analyze spend patterns and tipping behavior across different order types and geographic locations.
- **External Influences:** Evaluate the impact of weather, holidays, and inflation on sales performance.

---

2. Data Loading and Preprocessing

A robust analysis begins with the careful ingestion and cleaning of the dataset.

```
import pandas as pd

# Load transaction data
transactions = pd.read_csv("transactions.csv")
venues = pd.read_csv("venues.csv")

# Merge datasets on a common key
df = pd.merge(transactions, venues, on="venue_xref_id", how="left")

# Convert datetime fields
df['order_seated_at_local'] =
pd.to_datetime(df['order_seated_at_local'])
df['bill_paid_at_local'] = pd.to_datetime(df['bill_paid_at_local'])

# Handling missing values
df.fillna({"tip_amount": 0}, inplace=True)

print(df.head())
```

Data Sources:

- **Transaction Data:** Contains bill-level information including totals, taxes, payment details, timestamps, and waiter identifiers.
- **Venue Details:** Provides context such as city, restaurant concept, operating hours, and location identifiers.

Preprocessing Steps:

1. **Data Ingestion:**
    - Import CSV or database files into a unified DataFrame using libraries such as Pandas.
    - Merge datasets on a common key (e.g., `venue_xref_id`) to associate transaction details with venue characteristics.
2. **Data Cleaning:**
    - Handle missing values by either imputing (where appropriate) or filtering out incomplete records.
    - Ensure date and time fields (e.g., `order_seated_at_local`, `bill_paid_at_local`) are correctly parsed into datetime objects.
    - Normalize numeric fields such as bill totals, tip amounts, and order durations.
3. **Feature Engineering:**
    - Extract temporal features: hour of day, day-of-week, month, and flags for weekends or holidays.
    - Create derived metrics like tip percentage (i.e., `payment_total_tip` divided by `bill_total_billed`).
    - Construct a “start-of-day” offset feature to capture variations in venue operational hours.

```
# Extracting temporal features
df['hour_of_day'] = df['bill_paid_at_local'].dt.hour
df['day_of_week'] = df['bill_paid_at_local'].dt.day_name()
df['is_weekend'] = df['day_of_week'].isin(['Saturday', 'Sunday'])

# Calculating tip percentage
df['tip_percentage'] = df['payment_total_tip'] / df['bill_total_billed']
* 100
df['tip_percentage'].fillna(0, inplace=True)

print(df[['hour_of_day', 'day_of_week', 'is_weekend',
'tip_percentage']].head())
```

4. **Data Validation:**
    - Verify data consistency between merged tables.
    - Conduct exploratory checks to ensure feature distributions are as expected.

---

3. Exploratory Data Analysis (EDA)

Before diving into modeling, an in-depth EDA was conducted to understand the underlying patterns and distributions.

Key EDA Activities:

- **Descriptive Statistics:**
  - Summary statistics (mean, median, standard deviation) for sales, tips, and order durations.
  - Distribution plots (histograms, box plots) to identify outliers and skewness in key metrics.
- **Time-Series Trends:**
  - Line charts to visualize daily and hourly trends in order volume and revenue.
  - Seasonality analysis to detect recurring patterns, such as weekend spikes or holiday surges.
- **Correlation Analysis:**
  - Correlation matrices to identify relationships among variables such as operational offsets, tip percentages, and total revenue.
  - Scatter plots and heatmaps for visual correlation insights.
- **Segmentation:**
  - Grouping data by venue, order type (dine-in, takeout, delivery), and geographic location.
  - Initial clustering attempts to segment venues based on performance metrics.

These analyses provided the groundwork for selecting the appropriate models and methods in subsequent sections.

---

4. Problem Statements and Detailed Analyses

4.1. Peak Period Analysis & Staffing Recommendations

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

# Group data by hour to find peak times
hourly_orders = df.groupby("hour_of_day")["order_id"].count()
```

```
# Plot peak periods
plt.figure(figsize=(10, 5))
sns.lineplot(x=hourly_orders.index, y=hourly_orders.values, marker="o")
plt.xlabel("Hour of the Day")
plt.ylabel("Number of Orders")
plt.title("Peak Order Periods")
plt.xticks(range(24))
plt.grid()
plt.show()
```

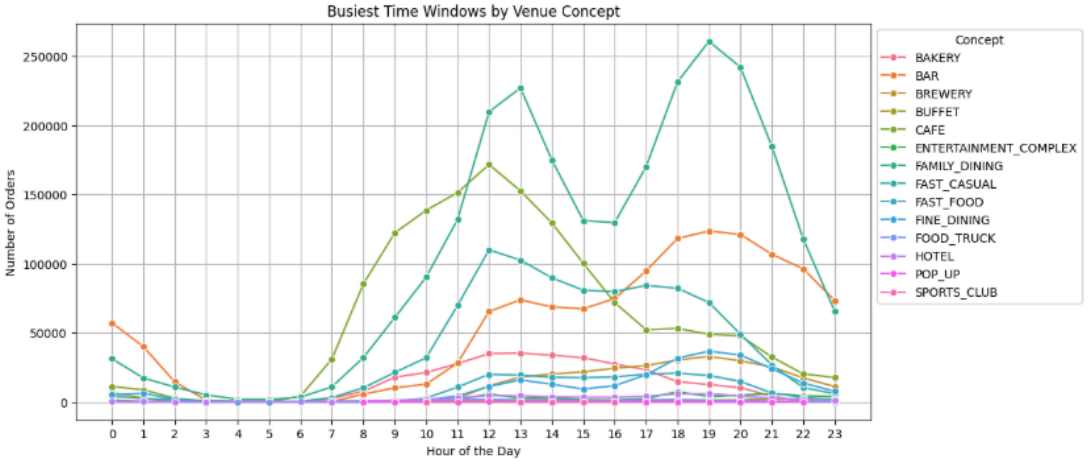
Objective:
Identify the busiest time windows across venues to propose staffing recommendations that improve service efficiency.

Methodology:

- **Time Extraction:**
  - Extract hour and day information from timestamp fields.
  - Create a new feature that maps each order to its corresponding hour slot.
- **Aggregation:**
  - Group data by venue and hour to calculate total orders and revenue.
  - Use pivot tables to compare peak versus off-peak hours.
- **Visualization:**
  - Line plots and heatmaps depict order volumes by hour.
  - Comparative charts between venues illustrate which locations experience the highest peaks.

Findings:

- The analysis highlights specific time slots where the order volume spikes.
- Recommendations include reallocating staffing during these peak hours and adjusting schedules dynamically based on historical trends.



-

4.2. Top Venues Analysis

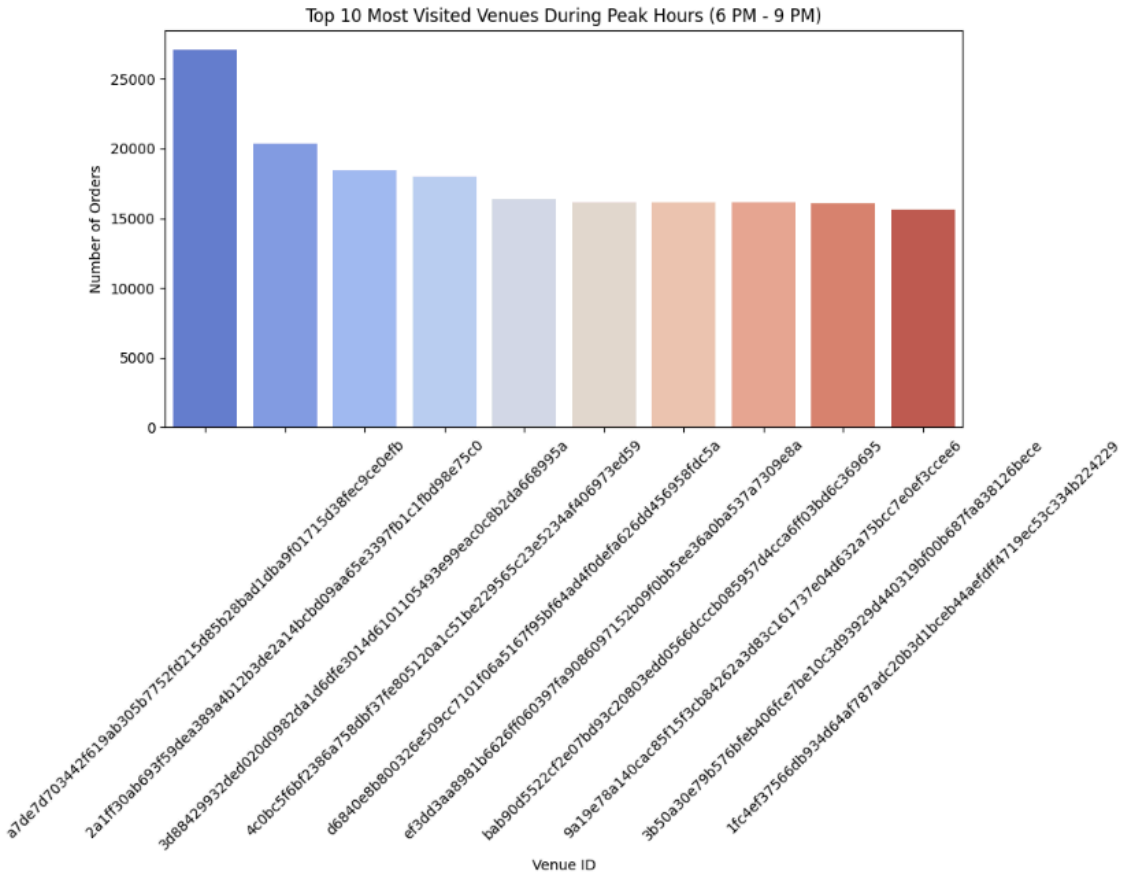
Objective:
Rank venues based on overall and peak period order volume.

Methodology:

- **Data Aggregation:**
  - Count total orders per venue.
  - Further refinement by isolating orders placed during identified peak hours (e.g., 6 PM to 9 PM).
- **Visualization:**
  - Bar charts to display the top 10 venues.
  - Dual comparisons showing overall performance versus peak-hour performance.

Insights:

- Some venues consistently perform well both overall and during peak times, suggesting strong market presence.
- Venues with high peak-hour performance might benefit from targeted promotions during off-peak periods.
-



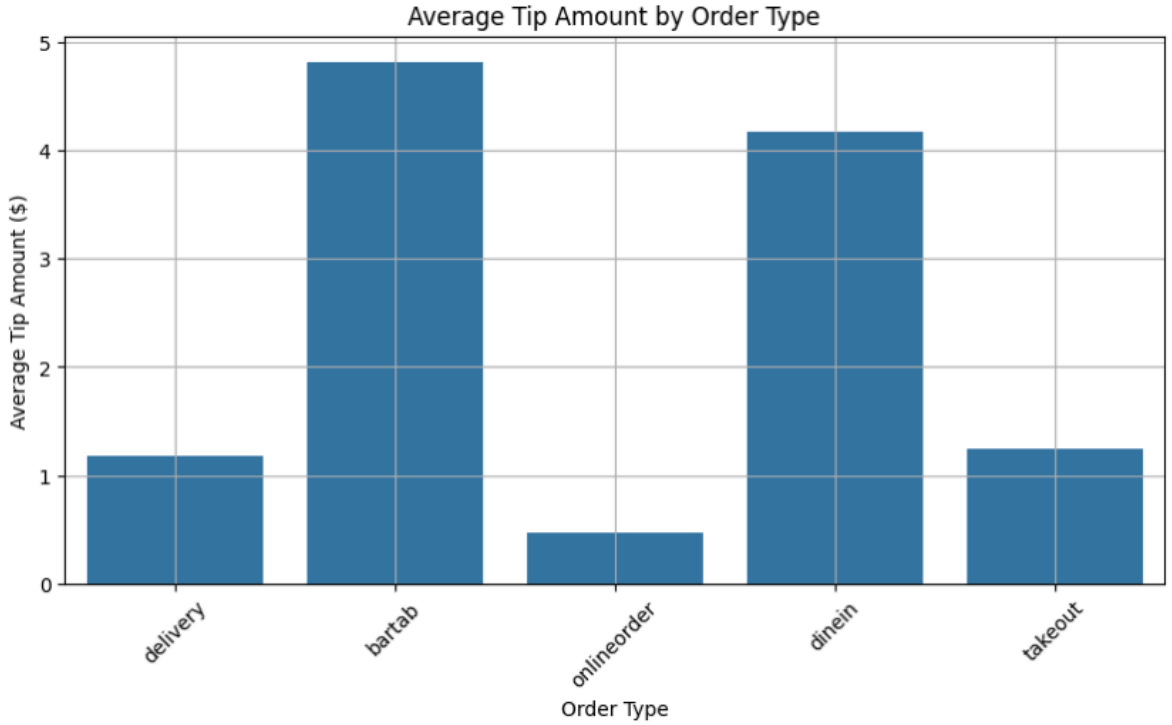
4.3. Customer Spend Insights

Objective:
Examine how order types impact customer spending, including bill size and tip behavior.

Methodology:

- **Grouping by Order Type:**
  - Divide data into categories (dine-in, takeout, delivery) using the `order_take_out_type_label`.
- **Metric Calculation:**
  - Compute average bill totals and tip amounts.
  - Calculate the tip percentage for each order.
- **Visualization:**
  - Bar charts and box plots compare average spending and tip percentages across order types.
  - Time-series plots may show variations in customer spending behavior over different periods.

Results:

- Dine-in orders might exhibit a higher average bill and tip percentage compared to takeout.
- Insights from this analysis could help tailor marketing strategies to encourage higher spend or improve tip practices.
- 

4.4. Forecasting Sales Revenue

Forecasting is critical for proactive decision-making. Two methods were explored: Prophet and Exponential Smoothing.

4.4.1. Forecasting with Prophet

```
from prophet import Prophet

# Prepare data for Prophet
sales_data =
df.groupby("bill_paid_at_local")["bill_total_billed"].sum().reset_index(
)
sales_data.columns = ["ds", "y"]

# Initialize and fit Prophet model
model = Prophet()
model.fit(sales_data)

# Make future predictions
future = model.make_future_dataframe(periods=30)
forecast = model.predict(future)

# Plot forecast
model.plot(forecast)
plt.title("Sales Forecasting with Prophet")
plt.show()
```

Objective:
Predict daily sales revenue for each venue using a Bayesian time series forecasting model.

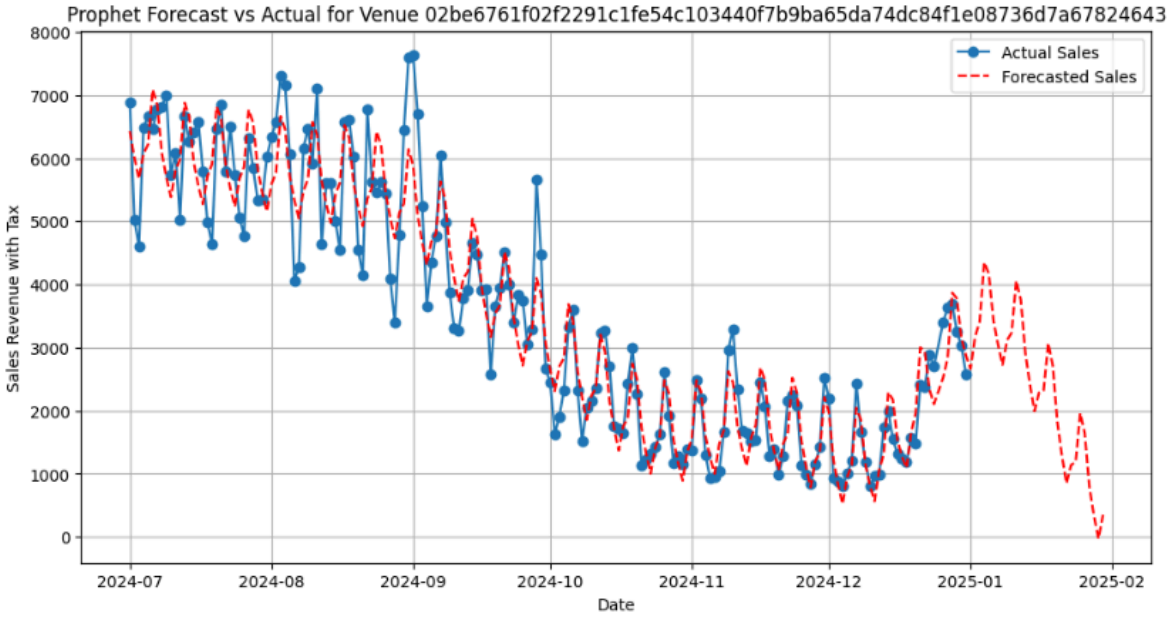
Methodology:

- **Data Preparation:**
  - Aggregate daily sales revenue per venue.
  - Format the dataset to meet Prophet's requirements (`ds` for date, `y` for revenue).
  - Incorporate additional regressors (e.g., weekend indicator, holiday flags).
- **Model Training:**
  - Loop over selected venues and fit individual Prophet models.
  - Forecast future revenue for a 30-day horizon.
  - Validate predictions against actual data using metrics such as Mean Absolute Error (MAE).
- **Challenges:**
  - Some venues had insufficient historical data, triggering errors like missing regressors.

- Fine-tuning was necessary to adjust seasonality parameters and handle outliers.

Outcomes:

- For venues with robust data, the Prophet model provided reasonable forecasts.
- The exercise illustrated the importance of data completeness and correct feature incorporation.



4.4.2. Forecasting with Exponential Smoothing

Objective:
Use Exponential Smoothing to forecast future sales revenue for selected venues.

Methodology:

- **Data Aggregation:**
  - Sum daily sales revenue and prepare the time series.
- **Model Specification:**
  - Fit the Exponential Smoothing model with both additive trend and seasonal components.
  - Define the seasonal period based on the data (e.g., weekly seasonality).
- **Error Handling:**
  - Address issues such as insufficient data span which may cause seasonal component errors.
  - Experiment with model parameters to achieve stable forecasts.

Results:

- When applied to venues with adequate historical data, Exponential Smoothing provided complementary forecasts.

- Comparisons between the Prophet and Exponential Smoothing forecasts highlighted the strengths and limitations of each method.

4.5. Effect of Operational Offsets on Sales

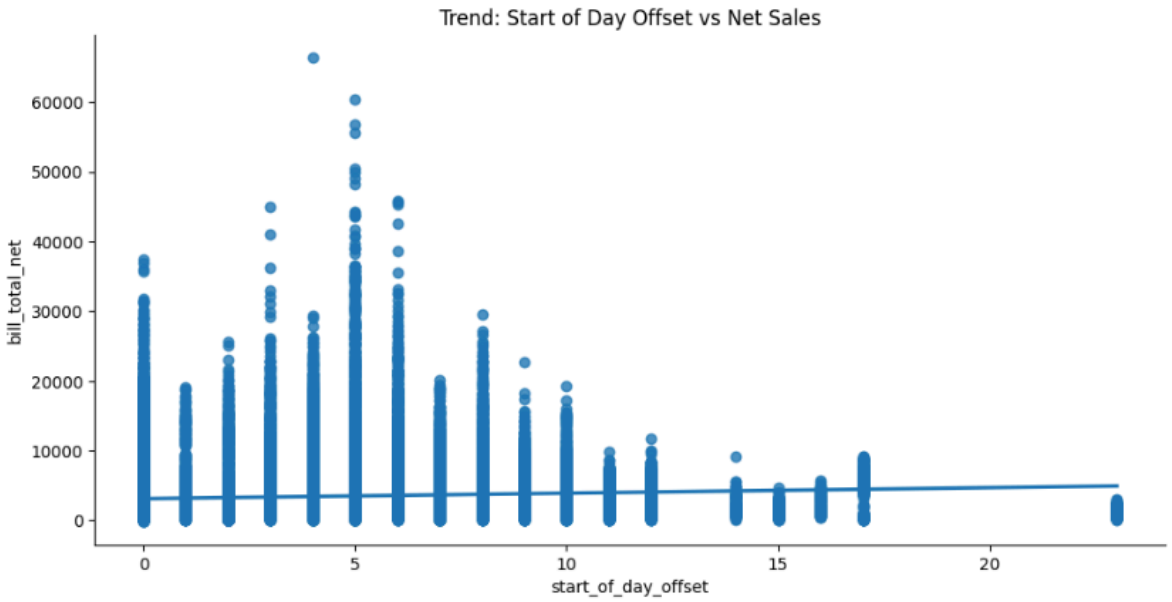
Objective:
Investigate the impact of the “start-of-day” offset on daily sales, tips, and net revenue.

Methodology:

- **Feature Extraction:**
  - Utilize the `start_of_day_offset` to group venues.
  - Aggregate daily performance metrics (sales, tips, net revenue).
- **Correlation Analysis:**
  - Compute correlation coefficients to examine the relationship between operational offsets and sales metrics.
  - Use scatter plots to visually inspect trends and outliers.

Findings:

- A low to moderate correlation suggests that while operational timing does play a role, other factors may be more dominant in influencing daily revenue.
- Further analysis could involve stratifying the data by venue type or region.



4.6. Waiter Performance Analysis

```
# Compute waiter performance metrics
waiter_performance = df.groupby("waiter_id").agg(
    avg_check_size=("bill_total_billed", "mean"),
    avg_tip_percentage=("tip_percentage", "mean"),
    total_orders=("order_id", "count")
).reset_index()

# Display top 5 waiters
print(waiter_performance.sort_values("avg_tip_percentage",
ascending=False).head())
```

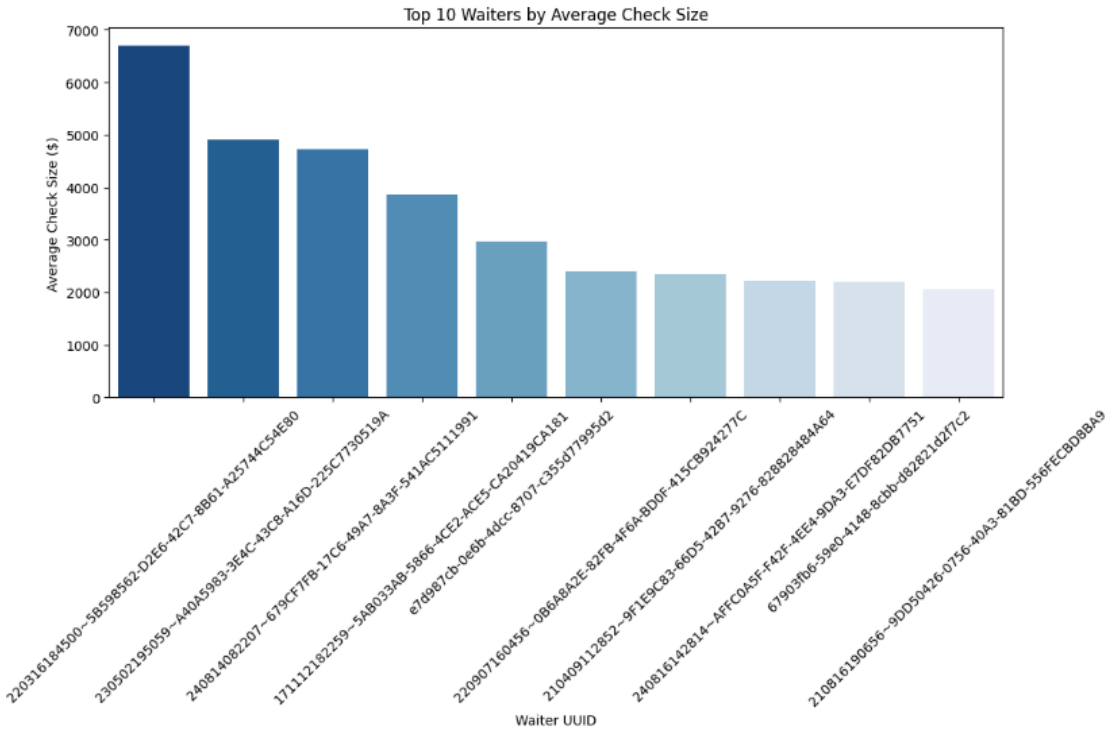
Objective:
Evaluate waiter performance based on average check sizes and tip percentages.

Methodology:

- **Data Filtering:**
  - Extract valid waiter identifiers and associated bill-level metrics.
- **Metric Computation:**
  - Calculate average check size and tip percentage per waiter.
  - Count the number of bills closed by each waiter.
- **Visualization and Ranking:**
  - Bar plots to rank waiters based on performance metrics.
  - Scatter plots to compare check sizes versus tip percentages.

Outcomes:

- The analysis surfaces top-performing waiters who consistently achieve high average bills and tip percentages.
- These insights could inform staff training or incentive programs.



4.7. Weather Data Correlation

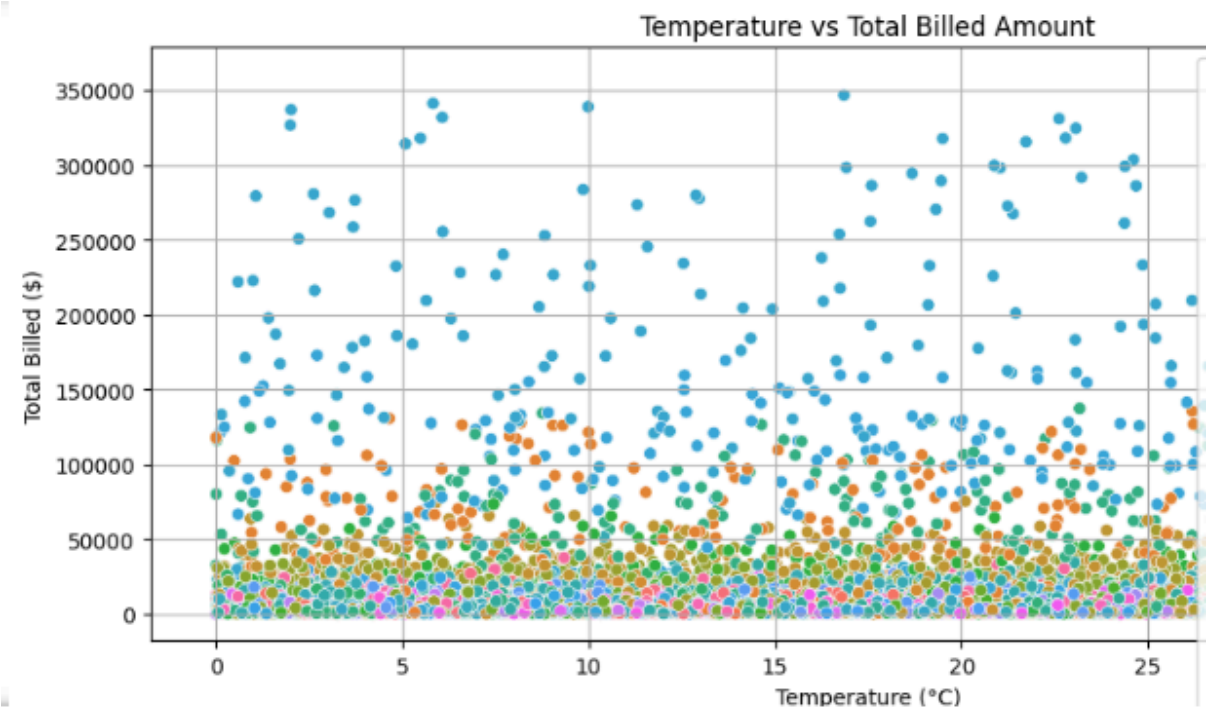
Objective:
Examine the influence of weather conditions (e.g., temperature, precipitation) on restaurant sales.

Methodology:

- **Data Simulation/Integration:**
  - For demonstration purposes, simulate or integrate external weather data for each venue's city and date.
- **Data Merging:**
  - Merge weather data with daily sales metrics.
- **Analytical Approach:**
  - Use scatter plots to assess relationships between weather variables and revenue or order volume.
  - Apply regression analysis to quantify the strength of these relationships.

Insights:

- Even simulated data reveals that extreme weather (e.g., heavy rain) might correlate with shifts in order type (more delivery orders) or overall revenue.
- With real weather data, this analysis could guide dynamic marketing or operational adjustments during inclement weather.



4.8. Holiday and Day-of-Week Impact on Sales

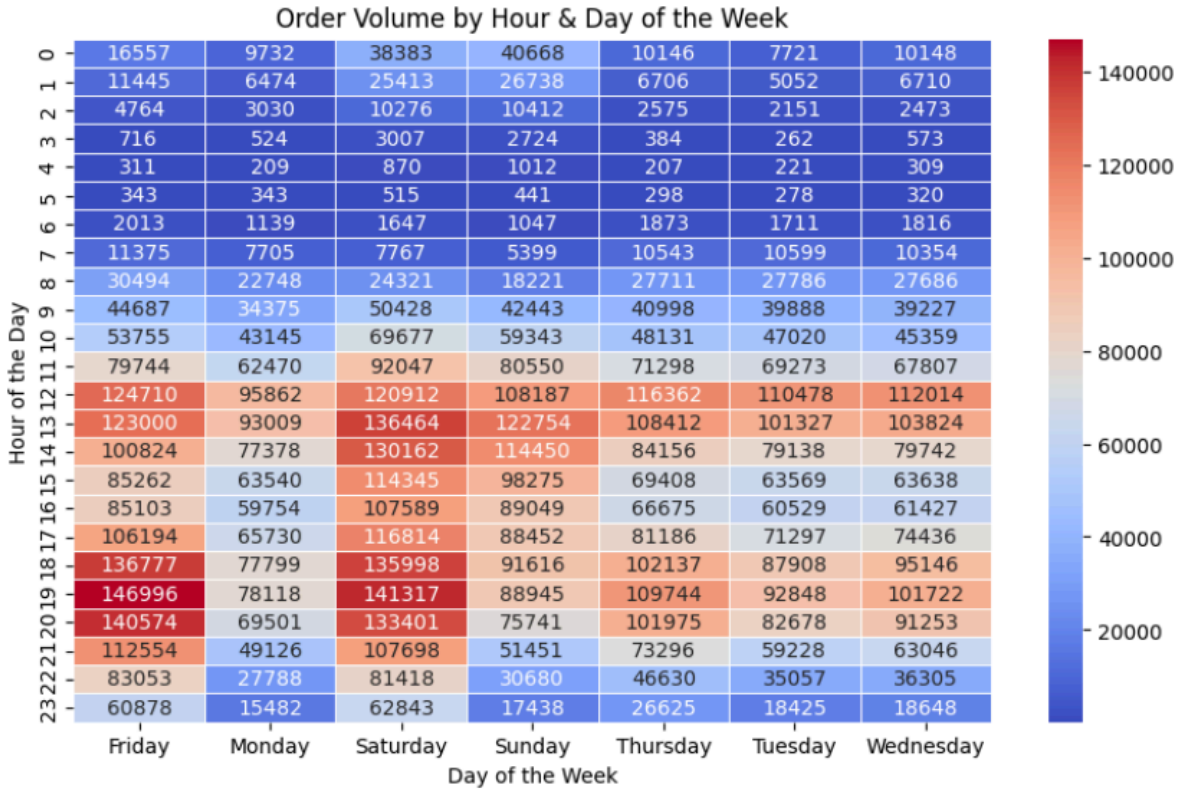
Objective:
Analyze how holidays and the day of the week influence overall sales and customer behavior.

Methodology:

- **Temporal Aggregation:**
  - Aggregate sales data by business date, day-of-week, and holiday status.
- **Visualization:**
  - Heatmaps and pivot tables illustrate variations in order volume and revenue.
  - Line plots and bar charts compare weekday versus weekend performance.

Findings:

- Certain holidays and weekends see a significant boost in order volumes, suggesting opportunities for targeted promotions.
- The analysis may also reveal patterns of decreased performance on specific weekdays, highlighting potential gaps in service or marketing.



4.9. Tip Culture by Geography

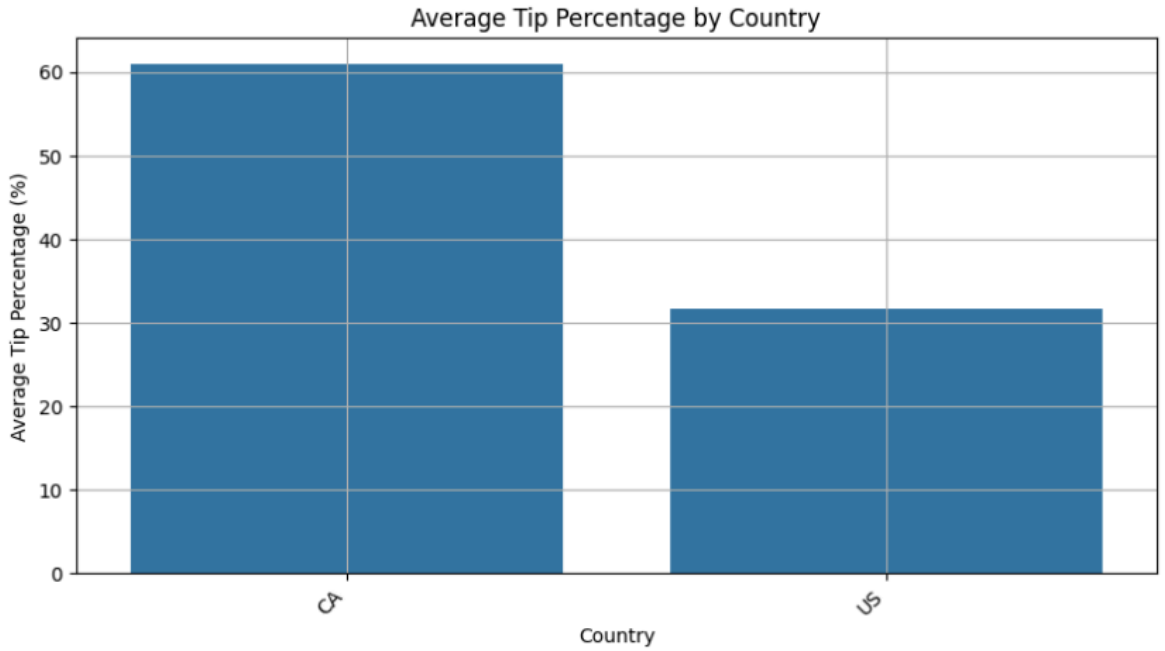
Objective:
Investigate regional differences in tipping behavior across cities and countries.

Methodology:

- **Calculation of Tip Metrics:**
  - Compute the tip percentage for each transaction.
- **Geographic Aggregation:**
  - Group data by city and country to derive average tip percentages.
- **Visualization:**
  - Bar charts and heatmaps to compare tipping trends across regions.
  - Statistical tests (e.g., ANOVA) may be employed to validate observed differences.

Insights:

- The analysis identifies areas with higher-than-average tipping, potentially reflecting cultural or service quality differences.
- These insights could support tailored staff training or region-specific service strategies.



4.10. Comparing Public Inflation Data and Sales

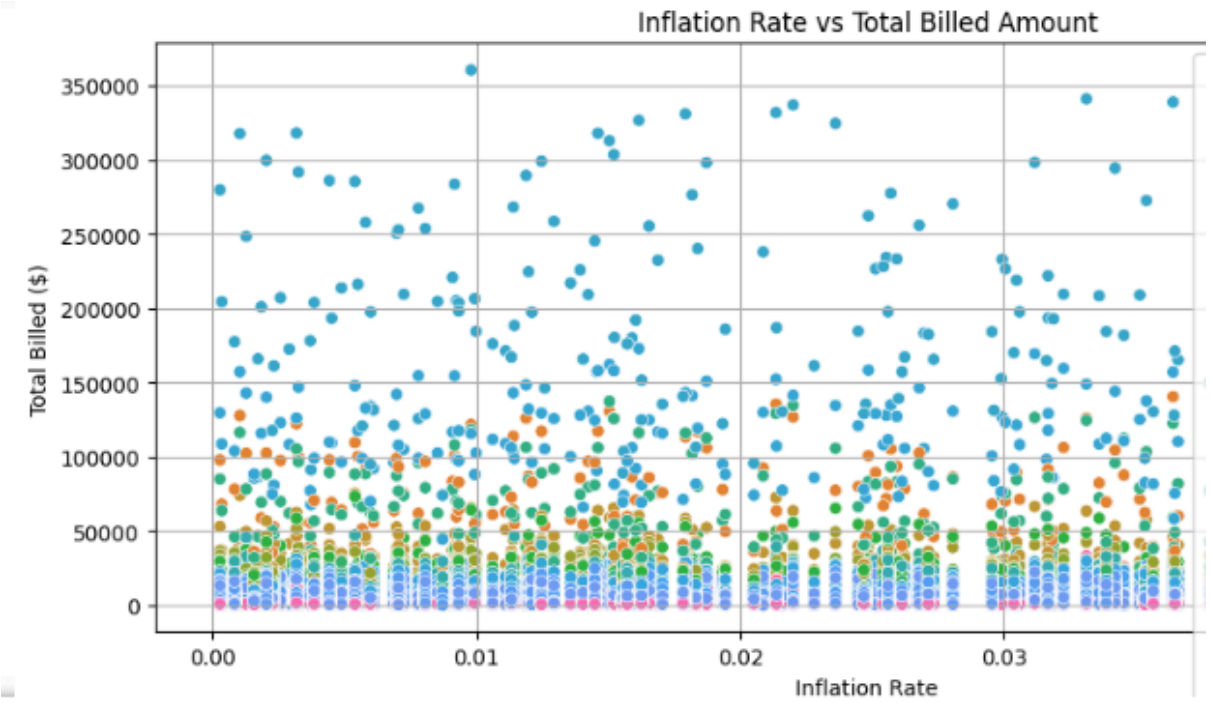
Objective:
Assess the correlation between inflation rates and restaurant sales to understand broader economic impacts.

Methodology:

- **Data Simulation/Integration:**
  - Simulate inflation data or integrate from publicly available sources.
- **Data Merging:**
  - Combine inflation data with daily sales metrics by matching on date and geographic region.
- **Analysis:**
  - Scatter plots and regression analyses are used to identify trends.
  - Investigate whether higher inflation corresponds with lower average bills or altered spending behavior.

Findings:

- While preliminary (with simulated data), results indicate that economic factors may influence consumer spending patterns.
- This analysis lays the groundwork for future work incorporating live economic data.



---

5. Conclusions and Future Work

Summary of Insights:

- **Operational Efficiency:** Detailed temporal analyses pinpoint peak periods and support dynamic staffing recommendations.
- **Revenue Forecasting:** Both Prophet and Exponential Smoothing models provided valuable forecasts, though challenges in data consistency underscore the need for comprehensive preprocessing.
- **Customer Behavior:** Significant differences in spending and tipping patterns were identified based on order type and geography.
- **External Factors:** Preliminary investigations into weather and inflation demonstrate the potential to integrate external data sources for a more holistic analysis.

Challenges and Limitations:

- **Data Quality:** Incomplete or sparse data for some venues limited the effectiveness of forecasting and clustering.
- **Model Tuning:** Several forecasting models required adjustments to handle seasonality and missing regressors.
- **External Data Integration:** Simulated data for weather and inflation highlights the need for real-world data feeds to improve model robustness.

Future Directions:

- **Enhanced Data Integration:** Incorporate real-time weather APIs, holiday calendars, and economic indicators.
- **Advanced Modeling:** Explore ensemble methods or deep learning approaches for improved forecasting accuracy.
- **User-Centric Insights:** Develop interactive dashboards to enable restaurant managers to dynamically explore these insights.

---

6. Appendices

Appendix A: Code Snippets

- **Data Loading and Cleaning:**
  Example snippet for reading CSV files and merging data.
- **Feature Engineering:**
  Code for extracting date-time features and calculating tip percentages.
- **Forecasting Models:**
  Excerpts from the Prophet and Exponential Smoothing implementations.

Appendix B: Additional Visualizations

- **Time-Series Plots:**
  Detailed plots of order volumes and revenue trends.
- **Heatmaps and Correlation Matrices:**
  Visual representations of the relationships between operational offsets, weather data, and sales metrics.

Appendix C: Model Evaluation Metrics

- Detailed tables of MAE, RMSE, and other performance metrics for each forecasting model.

---

7. References

- **TouchBistro Website:** Information on the service and regional presence.
- **Time Series Forecasting Literature:** Articles and documentation on Prophet and Exponential Smoothing methodologies.
- **External Data Sources:** (Planned for future work) Real-time weather APIs and economic datasets from government agencies.