Denver Coffee Shop Expansion

Hackathon Insights

March 9th, 2025

Kimhak Sou

Executive Summary

This report outlines the results of a data analysis project aimed at identifying the most strategic locations for a new coffee shop in Denver. The project was part of a Business Analytics Club Hackathon, focusing on areas with a high concentration of the target demographic (20-35 years old) and proximity to affluent households. The analysis leveraged geographical and demographic data to pinpoint the top three neighborhoods: University Neighborhood, North Capitol Hill Neighborhood, and Capitol Hill Neighborhood.

Introduction

The objective of this hackathon was to assist a Colorado-based coffee shop owner in expanding their business into Denver. The ideal locations were sought to be near affluent households and appealing to the 20–35-year-old demographic. The team collected and analyzed data on Starbucks locations in Denver and demographic information of Denver's neighborhoods to identify the best areas for expansion.

Data Sources and Methodology

Data Description

- Census Data: Provided demographic information for Denver's neighborhoods.
- Denver Data: Included additional insights specific to Denver's neighborhoods.
- Neighborhoods Coordinates: Geographical information to plot neighborhood locations on a map.

Methodology

The methodology involved cleaning the provided datasets, analyzing demographic proportions, and visually identifying the most promising neighborhoods for expansion.

Data Preparation and Cleaning

Given the datasets from the hackathon (census data, Denver data, and neighborhood coordinates), the data preparation and cleaning process will be as follows:

Step 1: Load the Data

```
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
from matplotlib.patches import FancyArrowPatch

#Load Denver Data
denver = pd.read_csv('denver.csv')
print(denver.head())

#Load Census Data
Census = pd.read_csv('Census .csv')
print(Census.head())

#Load Neighborhoods Data
Neighborhood = gpd.read_file('neighborhoods.shp')
print(Neighborhood.head())

#Check all the info and describe data
print(denver.info())
print(denver.describe())

print(Census.info())
print(Census.describe())

print(Neighborhood.info())
print(Neighborhood.describe())
```

- **Imports**: The script imports necessary libraries: pandas for data manipulation, geopandas for geospatial data, matplotlib.pyplot for plotting, and FancyArrowPatch for drawing arrows.

- **Data Loading**:
    - Loads Denver data from a CSV file into a pandas DataFrame.
    - Loads Census data from another CSV file into a pandas DataFrame.
    - Loads Neighborhoods data from a shapefile into a geopandas GeoDataFrame.

- **Data Inspection**:
    - Prints the first few rows of each dataset to get an initial overview.
    - Uses **info()** to display basic information about each dataset (e.g., data types, non-null values).
    - Uses **describe()** to generate descriptive statistics for each dataset (e.g., mean, standard deviation, quartiles).

Step 2: Inspect the Data

Inspect the datasets for missing values, incorrect data types, and inconsistencies.

```
# Check for missing values in each dataset
print("Missing values in census data:")
print(census.isnull().sum())

print("\nMissing values in Denver data:")
print(denver_data.isnull().sum())

print("\nMissing values in neighborhoods data:")
print(neighborhoods.isnull().sum())

# Remove duplicates from each dataset
census = census.drop_duplicates()
denver_data = denver_data.drop_duplicates()
neighborhoods = neighborhoods.drop_duplicates(subset='NBHD_ID')  # Assuming
'NBHD_ID' is the column for neighborhood IDs
```

- Check for Missing Values: The code uses the **isnull().sum()** method to identify and print the number of missing values in each column of the census, Denver, and neighborhoods datasets. This helps in understanding which columns have incomplete data that might affect the analysis.
- Remove Duplicates: The **drop_duplicates()** method is applied to each dataset to eliminate any duplicate rows. For the neighborhoods dataset, the subset parameter is used to specify that duplicates should be removed based on the 'NBHD_ID' column, ensuring that each neighborhood is represented only once.

Step 3: Analyze the Data

Calculate Demographic Proportions and High-income Household Proportions

```
# Calculate the proportion of the target demographic (18-34 years old)
Census['TARGET_DEMOGRAPHIC'] = Census['AGE_18_TO_34'] / Census['POPULATION_2010']

# Calculate the proportion of high-income households
Census['HIGH_INCOME_PROPORTION'] = Census['NUM_HHLD_100K+'] /
Census['NUM_HOUSEHOLDS']
```

- Calculate the proportion of the population within the 18-34 age range for each neighborhood. This is done by dividing the number of people in the target age group **(AGE_18_TO_34)** by the total population **(POPULATION_2010)** for each neighborhood.

- Calculate the proportion of households earning over $100,000 per year **(NUM_HHLD_100K+)** relative to the total number of households (**NUM_HOUSEHOLDS)** in each neighborhood.

Clean the Data Further

- After calculating the proportions, any rows with missing values in the **HIGH_INCOME_PROPORTION** column are dropped. This ensures that the subsequent analysis is based on complete data. The **.copy()** method is used to ensure that the changes are made on a separate copy of the original DataFrame, preserving the original data.

```
# Drop rows where 'HIGH_INCOME_PROPORTION' is NaN and create a copy
Census = Census.dropna(subset=['HIGH_INCOME_PROPORTION']).copy()
```

Step 4: Sort and Select Top Neighborhoods

- **Sorting Neighborhoods by Demographic Proportions:**
  - The code sorts the Census DataFrame by the **TARGET_DEMOGRAPHIC** and **HIGH_INCOME_PROPORTION** columns in descending order using **sorted_census = Census.sort_values(by=['TARGET_DEMOGRAPHIC', 'HIGH_INCOME_PROPORTION'], ascending=False).** This ensures that neighborhoods with the highest percentages of young adults (18-34 years old) and affluent households are ranked at the top, which is crucial for identifying locations with the most potential for a new coffee shop.

```
# Sort neighborhoods by the proportion of the target demographic and high-income
households
sorted_census = Census.sort_values(by=['TARGET_DEMOGRAPHIC',
'HIGH_INCOME_PROPORTION'], ascending=False)
```

- **Selecting the Top Three Neighborhoods:**
  - The code selects the top three entries from the sorted DataFrame with **top_three_neighborhoods = sorted_census.head(3).** These are the neighborhoods that have the highest proportions of the target demographic and high-income households, making them the most promising locations for attracting the desired clientele for the coffee shop expansion.

```
# Select the top three neighborhoods
top_three_neighborhoods = sorted_census.head(3)
print(top_three_neighborhoods[['NBHD_NAME', 'TARGET_DEMOGRAPHIC',
'HIGH_INCOME_PROPORTION']])
```

Step 5: Plotting the Visualization

- **Load and Merge Neighborhoods Data:**
  - The code begins by loading the neighborhood boundaries from a shapefile into a GeoDataFrame using **neighborhoods = gpd.read_file('Neighborhoods.shp').**
  - It then merges the top three neighborhoods identified in the previous analysis **(top_three_neighborhoods)** with the neighborhoods GeoDataFrame on the **NBHD_NAME** column, resulting in **top_three_gdf.**

```
# Load neighborhoods GeoDataFrame
neighborhoods = gpd.read_file('Neighborhoods.shp')

# Merge the top three neighborhoods with the neighborhoods GeoDataFrame
top_three_gdf = neighborhoods.merge(top_three_neighborhoods, on='NBHD_NAME')
```

- **Plot Neighborhoods and Highlight Top Three:**
  - A map is created with **fig, ax = plt.subplots(figsize=(12, 12)**) to provide a large visualization area.
  - All neighborhoods are plotted in light green with black edges using **neighborhoods.plot(ax=ax, color='lightgreen', edgecolor='black').**
  - The top three neighborhoods are highlighted in cyan with semi-transparent fill and black edges using **top_three_gdf.plot(ax=ax, color='cyan', edgecolor='black', alpha=0.7).**

```
# Plot neighborhoods and highlight the top three
fig, ax = plt.subplots(figsize=(12, 12))
neighborhoods.plot(ax=ax, color='lightgreen', edgecolor='black')
top_three_gdf.plot(ax=ax, color='cyan', edgecolor='black', alpha=0.7)
```

- **Plot Starbucks Locations and Annotate Top Neighborhoods:**
  - Starbucks locations are converted into a GeoDataFrame with **denver_gdf = gpd.GeoDataFrame(denver, geometry=gpd.points_from_xy(denver.Longitude, denver.Latitude))** and plotted in red.

```
# Plot Starbucks locations
denver_gdf = gpd.GeoDataFrame(
    denver,
    geometry=gpd.points_from_xy(denver.Longitude, denver.Latitude)
)
denver_gdf.plot(ax=ax, color='red', markersize=50, label='Starbucks Locations')

offsets = {
    0: (-20, -30),  # Moves up-left
    1: (-50, -30),  # Moves down-right
    2 : (50, 50)    # Moves down-left
}
arrow_Off = {

    0: (-0.02, -0.02),  # Moves up-left
    1: (-0.04, -0.02),  # Moves down-right
    2 : (0.04, 0.04)    # Moves down-left

}
index = 0
```

- ○ The top three neighborhoods are annotated with their names and arrows pointing to their centroids. This is done in a loop that iterates over the centroids of the top three neighborhoods, adding a text label and a FancyArrowPatch for each.

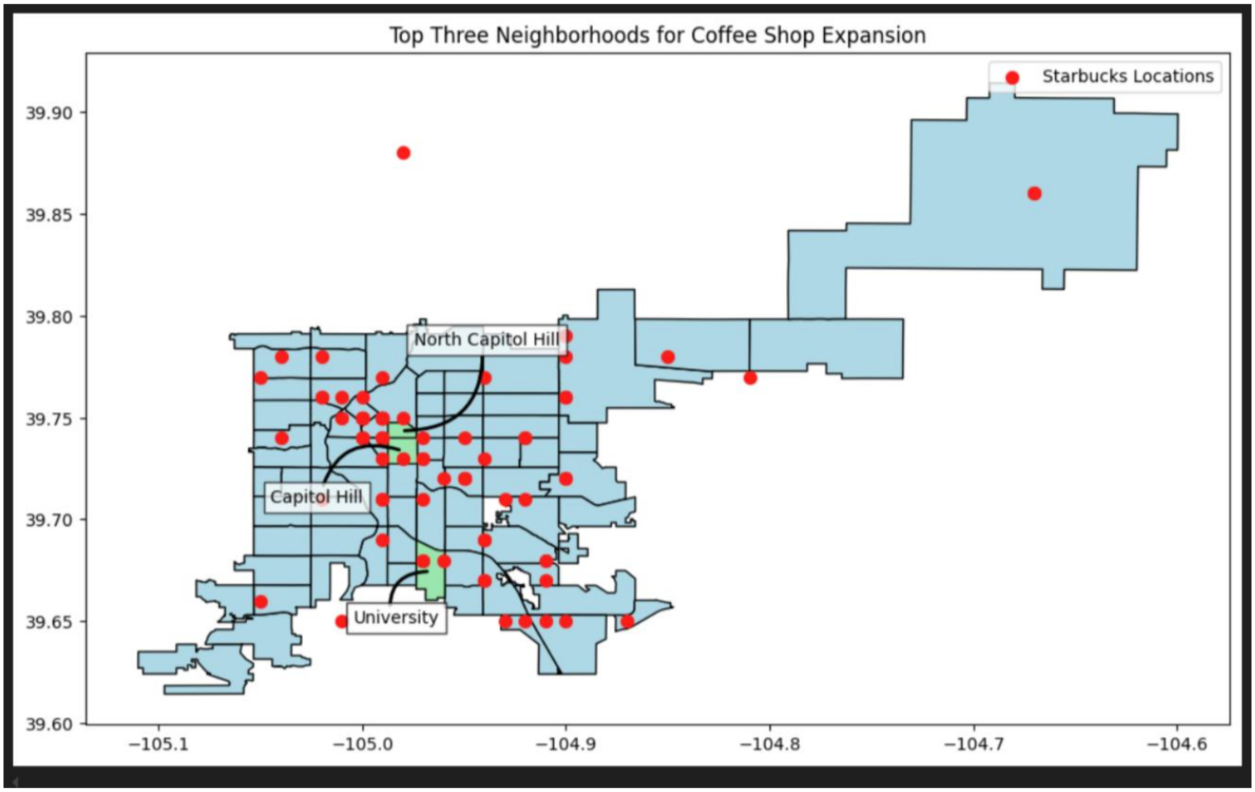
```
# Annotate the top three neighborhoods with their names
for x, y, label in zip(top_three_gdf.geometry.centroid.x,
top_three_gdf.geometry.centroid.y, top_three_gdf.NBHD_NAME):
    if pd.notna(label):  # Check for NaN values
        offset_x, offset_y = offsets.get(index, (10, -30))  # Default offset if
missing
        print(offset_x,offset_y,index)
        ax.annotate(label, xy=(x, y), xytext=(offset_x, offset_y),
textcoords='offset points',
                     ha='center', fontsize=10, color='black',
bbox=dict(facecolor='white', alpha=0.75))
        # Add an arrow pointing to the centroid

        arrow_x, arrow_y = arrow_Off.get(index, (10, -30))  # Default offset if
missing
        arrow = FancyArrowPatch((x, y), (x+arrow_x, y+arrow_y),
                                connectionstyle="arc3,rad=.5",
                                arrowstyle='->', color='black', lw=4)
        ax.add_patch(arrow) index += 1
```

- ○ The map is titled, a legend is added, and the plot is displayed with **plt.title('Top Three Neighborhoods for Coffee Shop Expansion'), plt.legend(), and plt.show().**

```
plt.title('Top Three Neighborhoods for Coffee Shop Expansion')
plt.legend()
plt.show()
```

Findings



Here are the neighborhoods that emerged as the top candidates for expansion:

1. **University Neighborhood**
   - **Target Demographic Proportion**: 61.7%
   - **High-Income Household Proportion**: 26.4%
   - This neighborhood stands out with the highest proportions of both the target demographic and high-income households, indicating a strong potential customer base with higher disposable income.
2. **North Capitol Hill Neighborhood**
   - **Target Demographic Proportion**: 55.6%
   - **High-Income Household Proportion**: 26.2%
   - It offers a strong demographic fit and comparable economic potential, making it a strategic secondary choice for expansion.
3. **Capitol Hill Neighborhood**
   - **Target Demographic Proportion**: 56.3%
   - **High-Income Household Proportion**: 11.3%
   - Despite having a lower proportion of high-income households compared to the other two, it still has a significant presence of the target demographic, making it a viable tertiary choice for expansion.

Recommendations

Based on the analysis, the following strategic recommendations are made for the expansion of the coffee shop into Denver:

- **Focus on the University Neighborhood**: Given its highest scores in both target demographic and high-income households, it should be the primary area for expansion efforts.
- **Consider North Capitol Hill Neighborhood**: With a strong demographic fit and comparable economic potential, it is recommended as a secondary area for expansion.
- **Evaluate Capitol Hill Neighborhood**: Its significant presence of the target demographic makes it a viable option, particularly if the business seeks to broaden its market reach.

Conclusion

The data-driven approach adopted in this analysis has laid a strategic foundation for expanding coffee shops in Denver. By targeting the University, North Capitol Hill, and Capitol Hill neighborhoods, the business can effectively reach a young, affluent clientele, maximizing success by aligning with desirable demographic and economic profiles.

Through this analysis, I gained valuable skills in data preparation, cleaning, analysis, and visualization. I improved my technical skills in Python (pandas, geopandas, matplotlib) and strengthened my critical thinking, report writing, and communication abilities. Managing this project enhanced my ability to draw insights and present data-driven recommendations effectively.

Note

This project has been a significant learning experience for me. If there are any inaccuracies or areas that could be improved, please feel free to let me know. I am eager to learn and continuously improve my skills.