1. Analyzing the Adjectives of City Wikipedia Pages: CSE 163 Final Project Report

Ian Hutchings

2. Research Questions

My research will focus on how different cities across the world are described in their Wikipedia pages by analyzing the most commonly used adjectives on each page. My research will be driven by 3 main research questions:

1. What are the most common adjectives used to describe all cities globally?
    a. The 20 most common adjectives are (in order):
    Local, new, former, important, early, administrative, small, municipal, old, square, urban, official, notable, modern, external, central, late, annual, economic, and industrial.
2. To what degree do the most commonly used adjectives change from city to city? How similar are the adjectives for cities in the same global region versus different regions?
    a. Overall cosine similarity: 0.11100
    b. Same continent cosine similarity: 0.13204
    c. Different continent cosine similarity: 0.10326
3. How accurately can a machine learning model predict the geographic location of a city based on the most common adjectives on its Wikipedia page?
    a. Regression analysis: Mean Absolute Error - Latitude: 11.9430
    Longitude: 28.8877
    b. Classification analysis: 0.69663 accuracy score

3. Motivation

In recent years, image-based machine learning (ML) models have gotten extremely accurate at identifying the geographical locations of Google Street View photos. From a single street-level image, they can often identify not just the correct country, but sometimes even the exact state or province where the photo was taken.

For this project, I wanted to shift the focus of ML geolocation from image-based to text-based data, seeing how well a machine learning model predicts the locations of cities using only text descriptions. I was curious about how much location-related information is encoded in the language used to describe a city, particularly the adjectives chosen to portray it.

4. Dataset

I drew from two data sources for this project. The first is the World Cities Database (https://simplemaps.com/data/world-cities). This is a csv file containing the name, country of residence, coordinates, and more information for over 40 thousand global cities. My second data source is Wikipedia. I used Wikipedia-API (https://pypi.org/project/Wikipedia-API/), a Python wrapper for the actual Wikipedia API that simplifies the data collection process. I collected the entire Wikipedia page text for each city in the World Cities Database for future analysis.

5. Method

Data Preparation:

- Loaded data from the World Cities Database and filtered it to only include relevant columns (city name, latitude, longitude, country, and country code (iso2)).
- Added a continent column for later analysis (RQ2) and removed cities with missing or invalid continents.
- Decreased the dataset size to a random sample of 20,000 cities to reduce excessively long future function runtimes.
- Added an adjectives column: a dictionary of the most common adjectives on the city's page capped at 50 adjectives, along with the number of times they appear. Adjectives were identified with SpaCy, a natural language processing library.
- Excluded adjectives that were either (1) too vague and common to provide meaningful information (e.g. "other" or "such"), or (2) explicitly referenced a country or global region (e.g. "Armenian" or "European"). The first category added noise to the data, harming the ability to find meaningful differences between cities. The second category introduced data leakage, directly revealing geographical information and allowing the model to "cheat". The list of all removed adjectives is available in invalid_adjectives.py.
- The pages the Wikipedia-API fetched were not always the correct city pages. For example, if I input the page name "Lebanon", it returned the page for the country, not the small town in Tennessee. To make sure the information for each city's page was accurate, I set a number of requirements that must be met before processing the page data. The requirements are:
    - The page must exist.
    - The page must have at least 1000 characters.
    - The name of the city's country must appear in the first 200 characters.
    - The terms "disambiguation" and "may refer to" must not appear in the first 200 characters.
    - The word "city" or "town" or "municipality" must appear in the first 200 characters.
- This filtered out extremely small pages and incorrect pages such as disambiguation pages, pages discussing a different topic under the same name, or pages for cities with the same name in different countries.

- Filtered the dataset to only include cities with at least 5 adjectives in their dictionary to ensure each city had a significant amount of data.
- Saved the final, cleaned dataframe to a csv file titled “wiki_cache.csv”.

Data Analysis:

- Mapped the locations of all cities in the dataset to check visually that the data was spread across every continent.
- Found the 20 most used adjectives across the entire dataset and plotted them in a pie chart. These results directly answered RQ1.
- Represented each city’s adjectives as a fixed-length vector (length = number of unique adjectives in dataset). Each index corresponded to an adjective, and each value was its count for that city’s page.
- Calculated cosine similarity between every pair of cities, and split into two groups: same-continent and different-continent. Computed mean similarity for each group. I ran a two-sample t-test to find the significance of my results. These results directly answered RQ2.
- Trained and tested a Random Forest Regressor model to predict the coordinates of a city based on its adjective dictionary (80/20 train/test split). I printed the model’s mean absolute error in latitude and longitude.
- Plotted a random set of 20 of the predictions the regression model made, drawing a line from the predicted location to the city’s actual location. This helped visualize the model’s accuracy.
- Trained and tested a Random Forest Classifier model to predict the continent a city resides in based on its adjective dictionary (80/20 train/test split). I printed the model’s accuracy and its classification report.
- The results of all of my ML analysis directly answered RQ3.

6. EDA

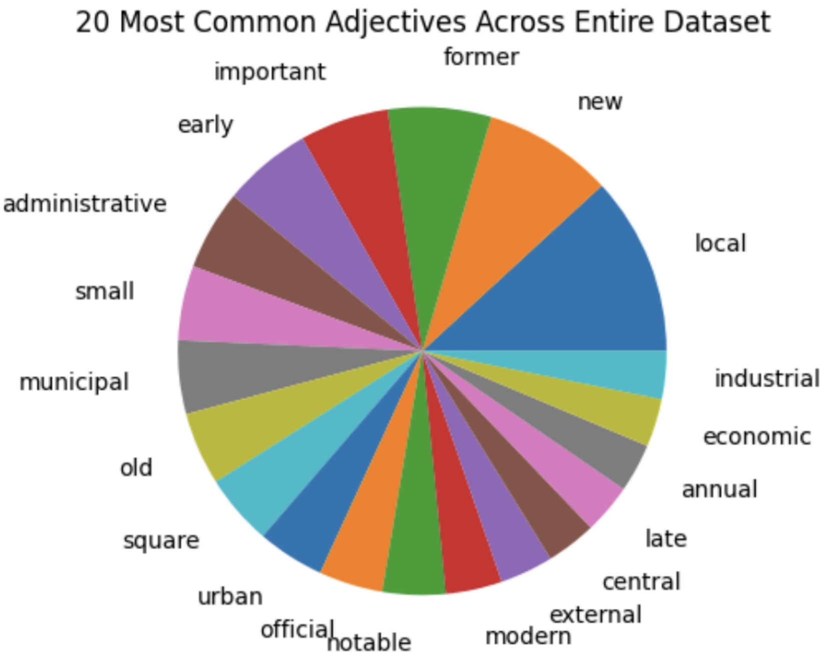
In the Exploratory Data Analysis I learned how to collect the Wikipedia data I needed and merge it with the global city database. I became familiar with what my data looked like and how it was distributed. That being said, most of the results I found during the EDA are irrelevant now, as those numbers represent the data before the rigorous cleaning and filtering I later implemented. For instance, in the EDA I reported that the dataset had 48060 rows and 7 columns. The final version of the data does have 7 columns, but only 5785 rows. It was important to note the data’s continental distribution. Notably, there are much fewer cities in Oceania than any other continent.

The parts of the EDA that prepared me the most for the rest of the project were identifying null and duplicate values. Since I retrieved the city data from a reputable online source, I didn’t spend much time considering how I would deal with messy data, until I wrote the EDA. Once I calculated the number of null values in the dataset, I realized that I would have to develop stringent conditions to handle null or inadequate values. While writing the EDA, I was also made to consider duplicate values, particularly cities with the same name. This consideration led me to write the strict page requirements I outline in the Methods section.

7. Results

RQ1: *"What are the most common adjectives used to describe all cities globally?"*

My findings for Research Question 1are displayed in the following pie chart.



Pie chart displaying the 20 most common adjectives across the dataset.

These adjectives do not include those listed in invalid_adjectives.py. As seen from the pie chart's distribution, the most common adjectives (MCAs) across the dataset are relatively evenly dispersed. It is arguable that a number of these MCAs do not provide much semantic information (e.g., "former", "annual") but as their prevalence is fairly low, they were not excluded. Understandably many of these MCAs reference history ("old", "new", "former") or a purpose that a city may serve ("administrative", "economic", "industrial").

RQ2: *"To what degree do the most commonly used adjectives change from city to city? How similar are the adjectives for cities in the same global region versus different regions?"*

To answer the first part of this question, I computed the cosine similarity between vectors representing the MCAs of every city in the dataset. I found that the average similarity score was 0.11098527786028889. To contextualize this value, all possible cosine similarity scores lie between 1 and -1. If two vectors are identical, they have a similarity score of 1. If they are orthogonal, the score is 0, and if they are exact opposites, the score is -1.

As the value that I calculated was close to 0, this means that the MCAs vary significantly across the entire dataset. This result was not entirely unexpected, though I was worried that the amount of noise presented by extremely common adjectives would make it difficult to differentiate the vectors and lead to a score closer to 1. I believe that the process of removing some of these common and vague adjectives likely significantly improved this score.

To answer the second part of this question, I split the cosine similarities into two groups: similarities between cities on the same and on different continents. I then calculated the mean of each group and the results are as follows: Average similarity (same continent): 0.13204158156566784. Average similarity (different continents): 0.10326488855998786.

I immediately saw that the similarity score for the same continent group was higher, but I wasn't sure how significant of a difference this was. To test the significance of my results, I conducted a two-sample t-test. My null hypothesis was that the two sample means were the same. My alternative hypothesis was that the two sample means were not the same. I used the standard threshold of p=0.05. I chose to use a t-test because the sample size included thousands of vectors, so under the Central Limit Theorem I could assume normality. The two samples were independent because there was no overlap between the two groups; every vector was either between same or different continent cities.

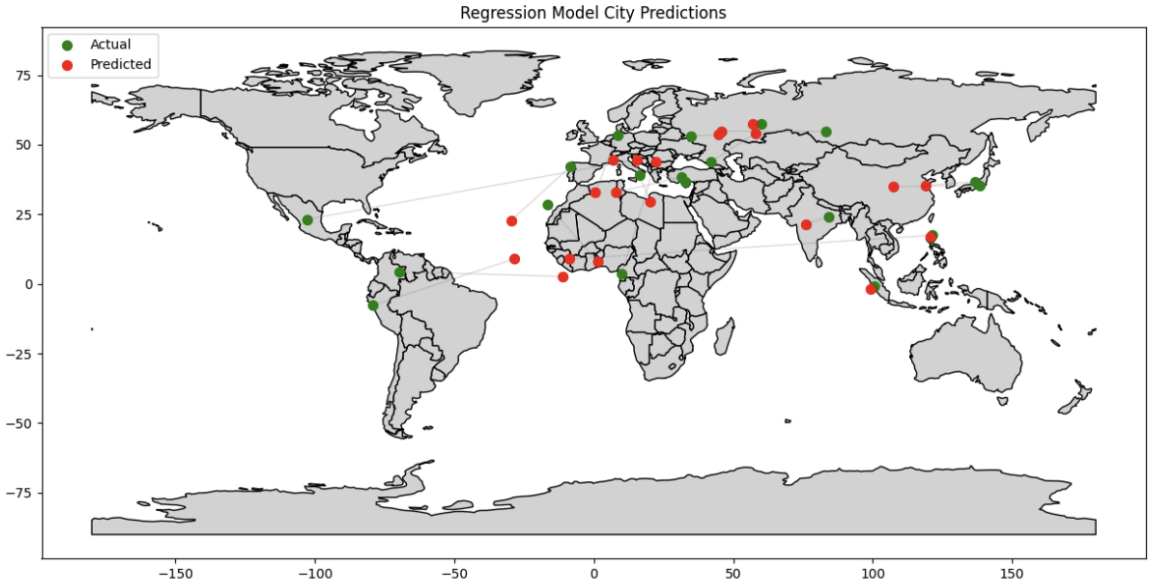
The results of this t-test were a T-statistic of 495.0336488019493 and a p-value of 0.0. The T-value represents the size of the difference relative to the variation in the sample. So, the higher the T-value is, the more significant the difference is. Coupled with the extremely low p-value of 0.0 (likely after some rounding), I can confidently conclude that the two sample means are significantly different.

These results were surprising to me. I was not expecting to obtain such a resounding answer. The results communicate that, at least on a continent-to-continent scale, the adjectives used in city Wikipedia pages contain a significant amount of locational information. Given how extreme the results are, there is cause to take it with a grain of salt, which is something I will discuss in more detail in the Limitations section. That being said, at the very least this was a sign that an ML model may be able to make some relatively accurate predictions.

***RQ3**: "How accurately can a machine learning model predict the geographic location of a city based on the most common adjectives on its Wikipedia page?"*

To answer this question, I first implemented a regression model to predict the coordinates of a city based on its MCA dictionary. The model performed with a mean absolute error of 11.943024158354254 degrees Latitude and 28.887682811802556 degrees Longitude. I chose to use mean absolute error as opposed to mean squared error as it is more robust to outliers, which are likely to occur with the large amount of data being analyzed.

For reference, these results are roughly equivalent to guessing in the western Saharan Desert for London, or near Baja California for Seattle. While these guesses are not exceptionally accurate, they are considerably better than throwing a random dart on a world map. To help visualize the model's accuracy, I plotted the predicted and actual locations of 20 random cities the model was tested on, as shown in the plot below.



World map plot showing the actual and predicted locations of 20 cities the regression model was tested on.

In this example, the model was able to guess nearly the exact location of a few cities (see Sumatra Island in Indonesia, western Russia, and India). These unique exceptional results may very well be explained by data leakage or other external factors, but the overall results are fairly impressive. The model was able to accurately predict at least the correct region of the world in most cases (except for two exceptionally bad guesses that missed Mexico and the Phillipines). Interestingly, in the northern Philippines the model produced both its most accurate prediction (nearly perfectly overlapping the true location) and its worst (placing the city in Côte d'Ivoire). This huge contrast is likely due to differences in the quality or quantity of information between the cities' Wikipedia pages, where one provided specific information to identify the exact location and the other did not.

It is important to note that every guess plotted was made in or near Afro-Eurasia. It appears as though the model's strategy was to guess near the center of the map in most cases, thus leading to its higher longitudinal error. The model does occasionally move from the center of the map, near Brazil or China, but the guesses do cluster around Europe and Northern Africa nevertheless.

After obtaining these results, I was interested in seeing how a different model would perform on what may be considered an easier task, identifying the continent of a city based on its MCAs. As the regression model was relatively accurate, I expected a classification model to do quite well.

The Random Forest Classifier tested with a classification accuracy of 69.66292134831461%. With nearly 70% accuracy, the model performed significantly well. If it guessed randomly each time, we would expect a result of roughly 17% (1 / 6 continents). The classification report showed the detailed statistics for each continent, as seen below.

```
Classification report:
            precision    recall  f1-score   support

      Africa       0.86      0.17      0.28       113
        Asia       0.70      0.72      0.71       347
      Europe       0.66      0.97      0.78       445
North America       0.93      0.51      0.66       131
     Oceania       0.00      0.00      0.00        19
South America       0.78      0.38      0.51       102

    accuracy                           0.70      1157
   macro avg       0.65      0.46      0.49      1157
 weighted avg       0.72      0.70      0.66      1157
```

Precision measures the accuracy of predictions for a continent, i.e. what percent of guesses for North America were actually in North America. Recall measures the model's ability to identify a continent, i.e. what percent of cities in Africa were correctly labeled as African. The model's guesses in North America, Africa, and South America were most precise, but they were also the least recalled. Notably, only 17% of all African cities and 38% of South American cities were identified. On the other hand, only 66% of Europe guesses were correct, but 97% of all European cities were correctly identified. Essentially, the classifier would repeatedly guess that a city is European and rarely guess outside of Europe or Asia. The model never guessed in Oceania, despite there being 19 cities in the continent. Balancing precision and recall (f1-score), the model performed best in Europe and Asia and worst in Oceania and Africa.

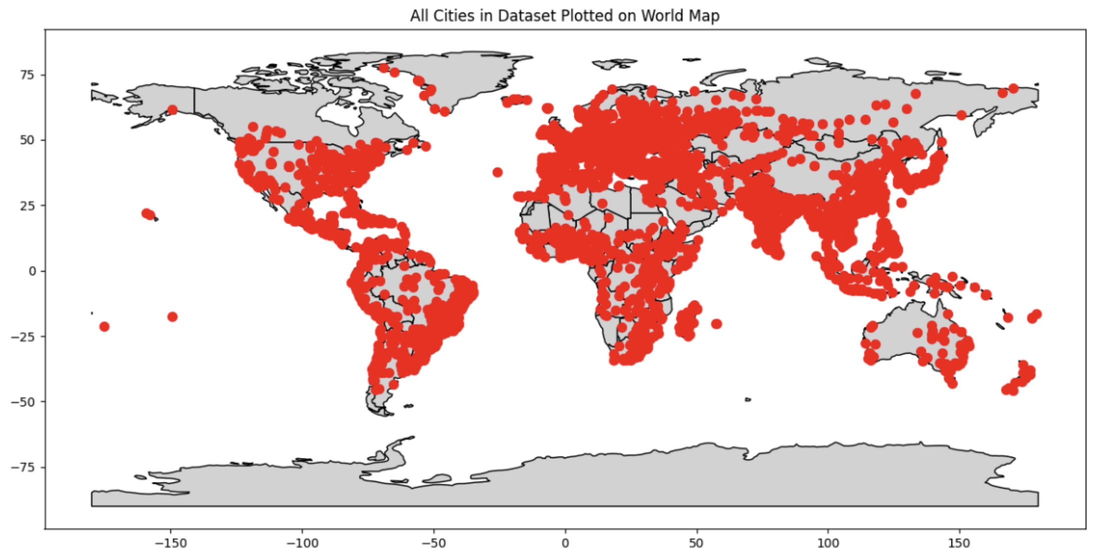
While the overall results are impressive, with an overall prediction accuracy of 70%, the continent breakdowns indicate a troubling pattern, which will be discussed further in the Limitations section.

8. Impact and Limitations

The first significant limitation of this report's findings is data bias. After finding each city's Wikipedia page MCAs and filtering out rows with inadequate data, the continent-by-continent break down of the dataset was as follows:

Europe: 2237
Asia: 1631
North America: 699
South America: 573
Africa: 559
Oceania: 85

The distribution is visualized in the following map:



World map plot showing the locations of all cities in the dataset.

Originally, the World Cities Database's distribution was more uniform, but after the data cleaning process, many cities in Asia, Africa, and South America were filtered out. This is most likely due to cities in the English speaking world having a higher quantity and quality of Wikipedia pages.

Despite the fact that Asia and Africa are the most populous continents on Earth, the cities in Europe and North America are more well documented on Wikipedia. This resulted in worse outcomes for the continents now with less data. As detailed in the Results section, the classification model was worse at predicting the continent of cities that were not in Europe or Asia. The skewed data can help explain the regression model's pattern of clustering guesses near Europe as well.

A second key consideration is the inherent messiness of the data and the inevitability of some data leakage. Language-based datasets are never perfectly clean, words can change meaning depending on context, and it is impossible to draw exact conclusions from the writing of strangers. Wikipedia strives for objectivity and neutrality but the text it contains is still subject to human error and bias.

The methods used in this project also introduce potential inaccuracies: an adjective might be misidentified, or a page, despite meeting all inclusion criteria, might still be matched to the wrong city. Given the scale of the dataset and the subjectivity of language, some errors are unavoidable. These can lead to data leakage, such as an invalid adjective slipping through because it is misspelled or appears in a different form, which could lead to overly-optimistic results. The filtering criteria (e.g., page text requirements, invalid adjective list) minimize these risks as much as possible, but they cannot eliminate them entirely.

Due to these limitations, the findings of my project should be taken with a grain of salt. While they present an interesting idea that I certainly think should be explored more, the results are likely prone to error and bias. In the future I hope to refine this project by using a more representative sample. I would be very interested to see how similar ML models would perform on different data sources, say rather than Wikipedia, a tourist/travel forum. The inherent differences in authors and language tone would likely produce significantly different results.

My conclusions should be used by others if they would like to build upon my work to create more air-tight results, or if they would simply enjoy experimenting with parameters or models and seeing how the results differ. They should not be used in any official capacity, but rather seen as an early exploration into a promising field of research.

9. Challenge Goals

Messy Data:

- Reasoning: I qualify for this challenge goal because I am using an API to get my data from Wikipedia.

Machine learning:

- A large part of my research project involved training and testing two different ML models. I used a RandomForestClassifier and RandomForestRegressor.
- I changed this challenge goal slightly, as I originally planned to only use one model, but with different hyperparameters. I chose to instead use two different models (a regressor and classifier), as I thought the results would be more interesting and cover a larger scope.

10. Work Plan Evaluation

Original work plan:

1. Collect and properly format data (4-6 hours)
2. Compute preliminary data analysis and answer RQ1 and RQ2 (6-8 hours)
3. Train and test LM model to answer RQ3 (6-10 hours)

It's difficult to say as I didn't track hours of work, but I think I spent much more time than I anticipated. The upper ends of these ranges were relatively accurate, but I didn't account for the time it took to write testing functions, add comments and attribute sources, write the report, and several other small details. If I fully understood what the project involved from the beginning, I think my estimates would have been more accurate.

11. Testing

I tested my code using a number of different methods. For every non-ML function, I tested with a smaller dataset, asserting that the values that I know to be true are correctly returned. For example, I checked that the country_to_continent function would return South America for Brazil.

For the machine learning functions, I checked that the results contained the correct information, e.g. scores for latitudinal and longitudinal error for the regression_analysis function. I also tested the models against a minimum threshold of accuracy, e.g. that classification_analysis has an accuracy score of at least 25%, as otherwise the model would essentially be guessing randomly.

To test my plotting functions, I ensured that the number of points plotted matched the number of cities that should be plotted.

12. Collaboration

All work for this project was done by me. All the sources I consulted are included in the function comment where they were used.