DATA 512: Visualizations and Reflection

Ananya Bajaj

1) **Produce a histogram showing the number of fires occurring every 50 mile distance from your assigned city up to the max specified distance.**

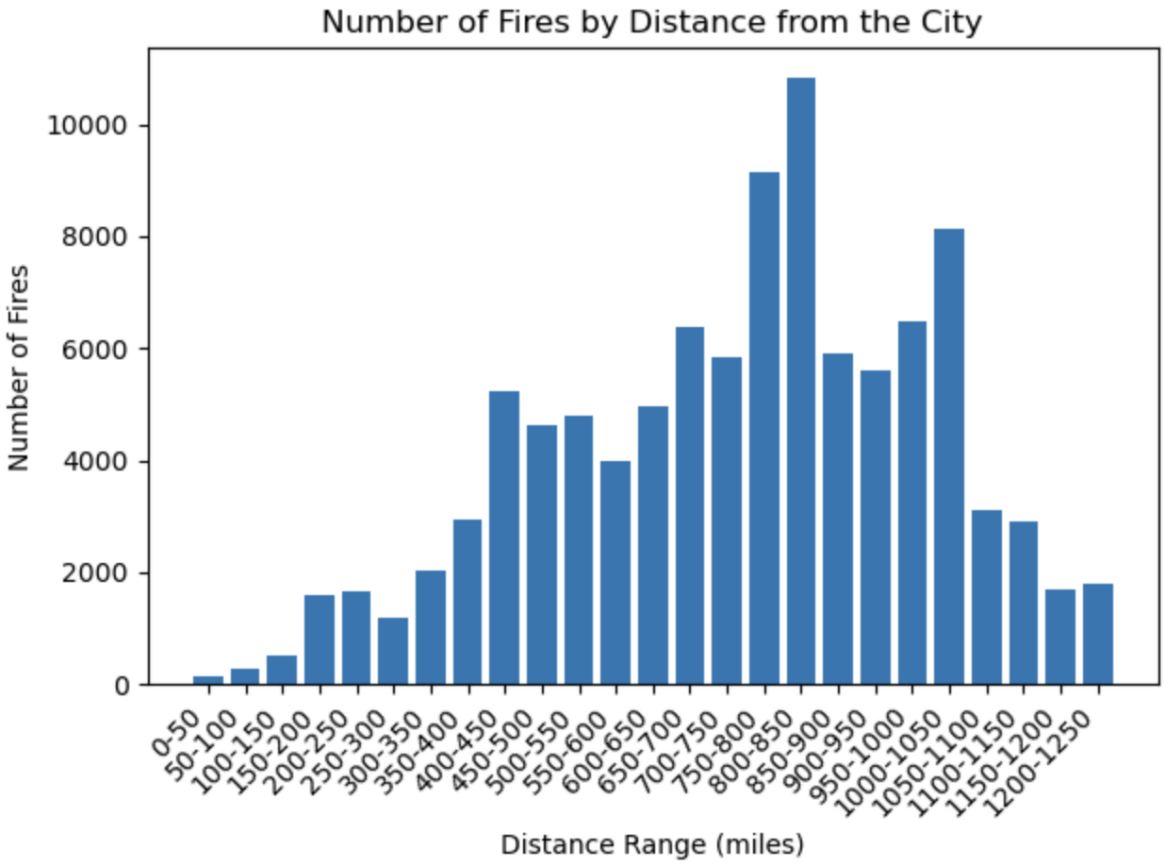


Figure Description: This bar chart illustrates the distribution of wildfires based on their proximity to the designated city, Pueblo West. The horizontal axis represents distance in miles, divided into 50-mile intervals, while the vertical axis displays the number of fires within each distance category. Upon examination, it is evident that the highest concentration of fires around Pueblo West occurs within the 800-850 mile range.

Interpretation: To interpret the chart, start from the left side of the distance axis, representing the distance from Pueblo West in miles. Each bar corresponds to a specific distance range (e.g., the first bar covers fires within 0-50 miles). The height of each bar indicates the quantity of fires within that distance range. This visualization

offers insights into the spatial distribution of wildfires in relation to Pueblo West, with peaks suggesting significant fire occurrences at specific distances from the city.

Underlying Data and Processing: The analysis utilizes wildfire data to assess the relationship between fire size and distance. It tallies the number of fires within different distance categories from a particular city. The underlying data comprises wildfire features, including attributes like fire year, GIS acres burned, fire type, and geometry data representing the fire's perimeter.

Smoke Impact Estimation: The calculate_smoke_impact function estimates smoke impact based on fire size, distance from the city, and fire type, using the formula: weight * ((fire_size)**2 / (distance))

Annual Smoke Impact Estimates: An annual_smoke_impact dictionary is established to store smoke impact estimates. For each wildfire feature meeting specific conditions (e.g., within the last 60 years, during the fire season), the smoke impact is calculated and added to the corresponding year's list in annual_smoke_impact.

Fire Counts and Distance Ranges: A fire_counts dictionary is created to track the number of fires within designated distance ranges from the city (in 50-mile intervals). For each wildfire feature, the distance from the city is computed, and the corresponding distance range is identified. If the distance falls within the predefined range (up to 1250 miles), the count for that range is incremented.

Time Tracking: The code incorporates time tracking using the time module to measure the processing time for features.

2. Produce a time series graph of total acres burned per year for the fires occurring in the specified distance from your city.

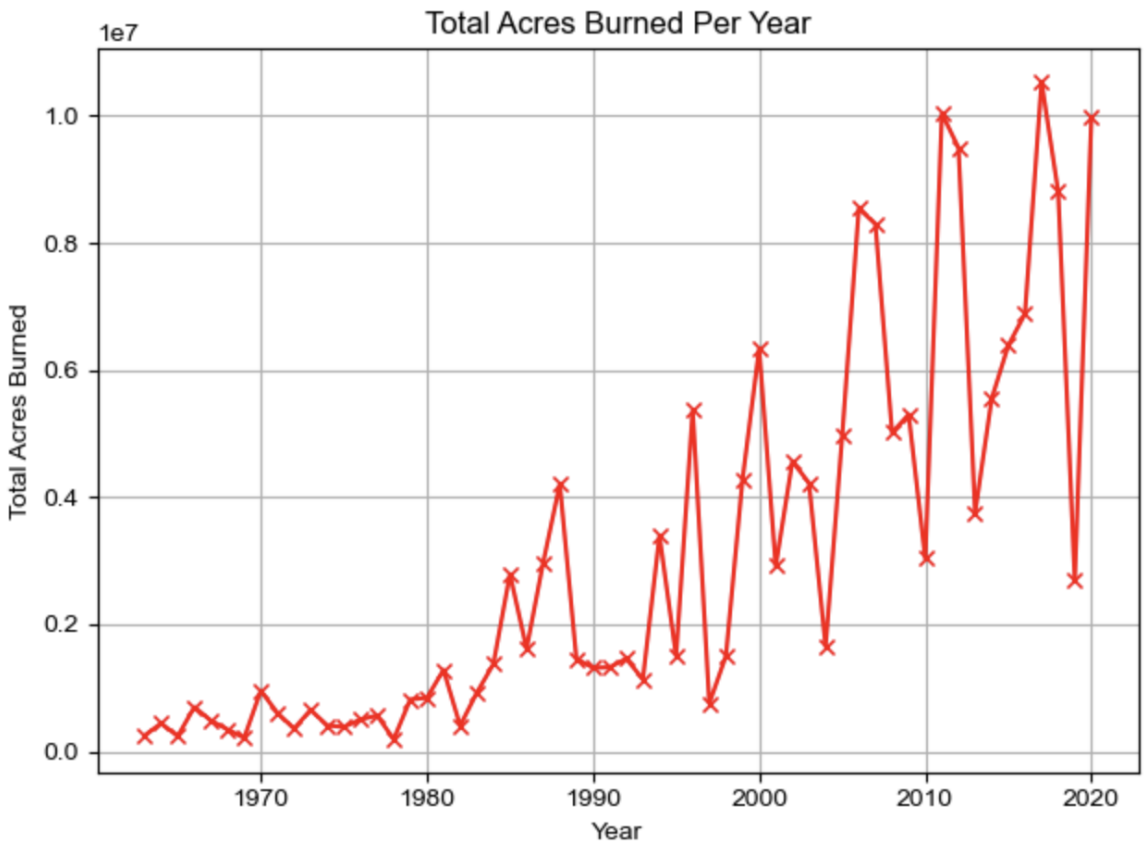


Figure Description: This time series chart illustrates the annual total acres burned by wildfires within a specified distance from Pueblo West. The horizontal axis denotes the years, and the vertical axis indicates the total acres burned (scaled by multiplying the values by 10,000,000). It is evident from the data that the cumulative acres burned around my assigned city, Pueblo West, have increased over time.

Interpretation: To interpret this graph, examine the x-axis representing years and the y-axis displaying the total acres burned. Each data point on the chart corresponds to a specific year, revealing the cumulative acres burned for wildfires within the designated distance from Pueblo West. This visualization offers insights into the temporal trends and patterns of wildfire activity in terms of total acres burned per year within the specified radius from Pueblo West.

Underlying Data and Processing: The data used for this visualization includes the following variables:
wf_feature['attributes']['Fire_Year']: The year of the wildfire event.
wf_feature['attributes']['GIS_Acres']: The acres burned by the wildfire.

wf_feature['attributes']['Assigned_Fire_Type']: The assigned type or category of the wildfire.
wf_feature['geometry']['rings'][0]: The geometry data representing the perimeter of the wildfire.

Data processing involves the following steps:

Iterating Through Wildfire Features: The code iterates through a list of wildfire features (feature_list) to access information about each individual wildfire event.

Filtering by Year and Fire Season: For each wildfire feature, the code checks if the fire year (wf_year) falls within a specified range (between 1963 and 2023) and if it occurred during the fire season.

Calculating Distance from City: The code utilizes the function

average_distance_from_place_to_fire_perimeter to compute the shortest distance from the specified city (Pueblo West) to the perimeter of the wildfire.

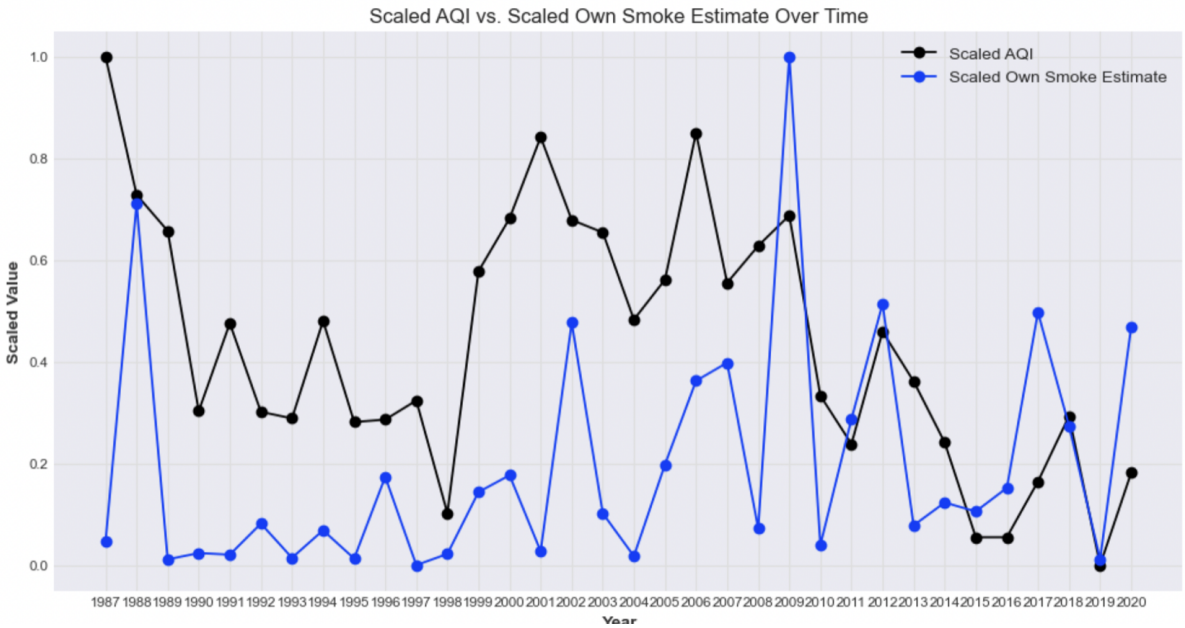
Checking Distance Threshold: It ensures that the distance to the fire is within 1250 miles, as specified in the code.

Calculating Smoke Impact: The calculate_smoke_impact function estimates the smoke impact based on fire size, distance, and fire type, using the formula smoke_impact = (fire_size)^2 / distance.

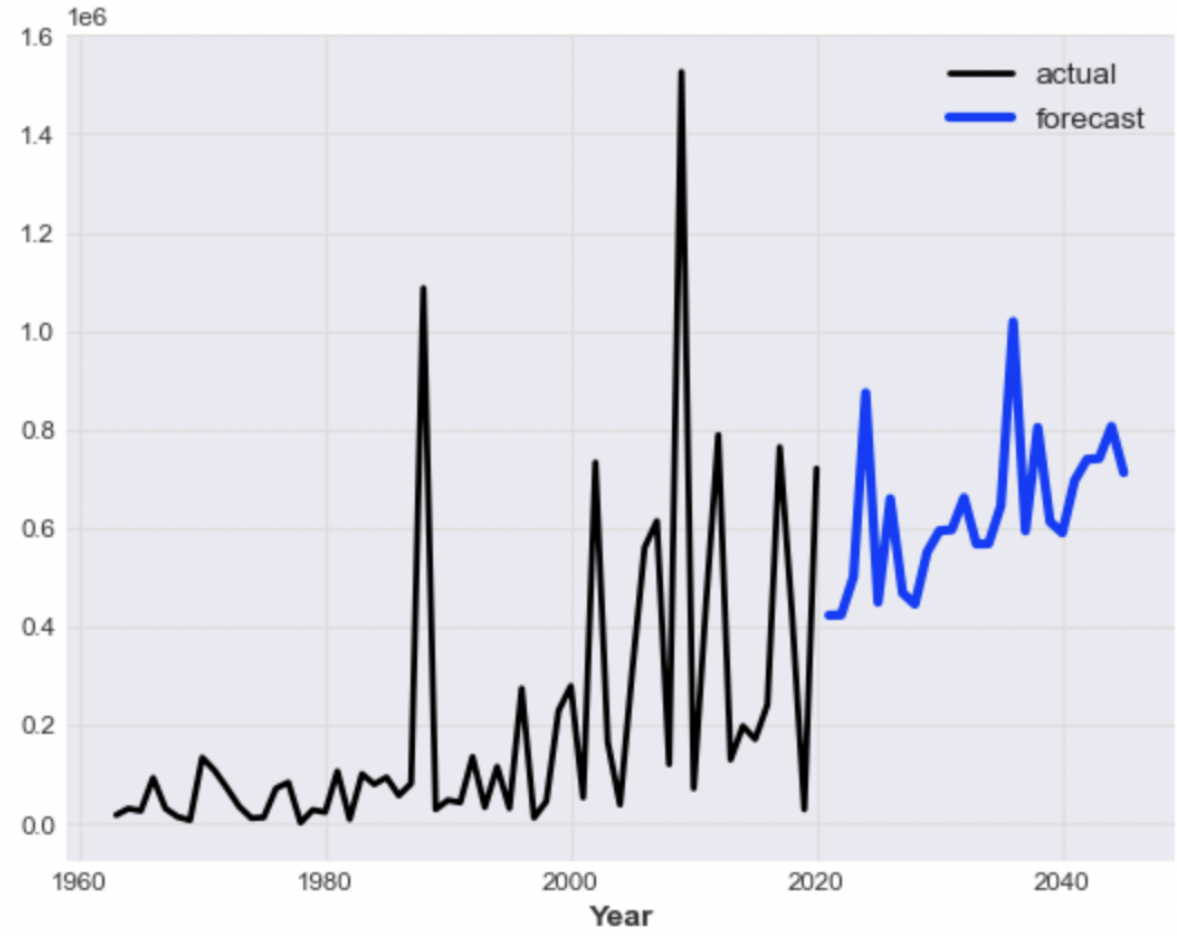
Aggregating Data for Visualization: The code maintains two main data structures:
annual_smoke_impact: A dictionary holding lists of smoke impacts for each year. If the year is not present, a new entry is created; otherwise, the impact value is appended to the existing list.

acres_burned_annual: A dictionary tracking the total acres burned per year. If the year is not present, a new entry is created with the acres burned value; otherwise, the value is added to the existing total.

3) Produce a time series graph containing your fire smoke estimate for your city and the AQI estimate for your city.



Time Series Forecasting



I used Darts - the Data Analytics and Retrieval Toolkit, which is a Python library specializing in time series forecasting. Within Darts, I employed the exponential smoothing model. The model utilizes exponentially decreasing weights for past observations, giving more prominence to recent data points while diminishing the impact of older ones. The library offers a user-friendly interface for model initialization, training, forecasting, and hyperparameter tuning. Exponential smoothing models, known for capturing trends and seasonality in time series data, can be easily evaluated using metrics such as mean absolute error (MAE) and mean squared error (MSE) through Darts' comprehensive set of functionalities. I conducted a forecast for the next 25 years as depicted in the chart above.

Reflection

This assignment facilitated collaborative work, proving to be highly beneficial for us. It offered a broader perspective on the problem and diverse approaches to its solution. Through the exchange of code snippets, statistical methods, and visualization techniques, I gained insights from my classmates, enhancing efficiency and productivity. Instead of struggling with specific aspects, I could engage in discussions with peers, learning about their approaches and problem-solving strategies. This diversity was valuable, as different classmates brought unique methods to the table. Observing alternative approaches contributed to a more thorough understanding of the material.

Collaboration also provided opportunities for validation and clarification. When uncertain about a particular code snippet or statistical approach, discussing it with classmates ensured correctness. Upon completing this project, I can confidently assert that collaboration nurtures a sense of community and shared learning. It promotes open communication, knowledge exchange, and a supportive environment for all students to thrive. My positive experience with collaboration reinforces my belief that it is the key to the successful completion of projects.

How did the possibility of collaboration help, hinder, or change your thinking about the problem? Your reflection statement should include specific attributions for all code, methods, and techniques that you reused. Your reflection statement should be no more than two written pages.

Participating in the collaborative activities for this assignment proved to be a transformative experience, shaping my understanding and approach to problem-solving in significant ways.

- While I spent most of my time on the code, discussing various bugs and errors faced by other students gave me a broad perspective, and tried to overcome those.
- It really helped me troubleshoot the problems that came my way while working on the code.
- As for hindrance, I feel that since different students had different cities, the gaseous and particulate part was kind of confusing. Since even after running the loop it was frustrating when the values were missing in between, and so asking other students did not help either as to what numbers they were getting.
- This project changed my thinking substantially by really figuring out how to run APIs and document code properly.

Collaboration acted as a catalyst for efficient problem resolution. Instead of navigating roadblocks in isolation, I leveraged the collective knowledge and experiences of my classmates. Discussing challenges with them provided valuable insights into tackling specific problem areas. This collaborative problem-solving approach significantly increased productivity.

Attributions for code, methods, and techniques.

I used the API methods provided by the professor in the code workbook, I mostly used that to tweak my solutions. I also discussed with 1-2 classmates about the time series forecasting methods that they used and compared the results.