CNC Part Cost Estimation and Supplier Matching

Data Collection via Web Scraping:

Methodology:

1. Introduction: The methodology employed for web scraping and data collection involved extracting information from a web page displaying product cards on the Indiamart directory. This process aimed to gather supplier details, including names, locations, materials used, and price ranges for CNC machine components. The approach utilized Python's BeautifulSoup library for parsing HTML content and requests module for fetching web pages. The collected data was structured into a dictionary format and subsequently transformed into a Pandas DataFrame for further analysis and export into CSV format.

2. Data Collection Process: To initiate the data collection, the script began by accessing the Indiamart search page specifically tailored for CNC machine components. Using BeautifulSoup, all links associated with product cards ('cardlinks') on the page were identified and stored in a list (`links_list`). Each link represented a specific supplier's page containing detailed information about their offerings.

3. Iterative Data Retrieval: Subsequently, the script iterated through each link in `links_list`. For each iteration, it attempted to access the supplier's page, retrieve relevant data such as supplier name, location, materials used, and price details using appropriate HTML tags and attributes. Error handling mechanisms were implemented to manage cases where links redirected unexpectedly or data was missing, ensuring the script continued execution without interruption.

4. Data Structuring and Conversion: Collected data was stored in a dictionary (`d`) with keys representing different attributes (Supplier_Name, Location, Materials, Price). This structured approach facilitated systematic storage and management of retrieved information. Upon completion of data collection, the dictionary was converted into a Pandas DataFrame (`df1`), leveraging the powerful data manipulation capabilities of Pandas for further processing and analysis.

5. Exporting Data: The final step involved exporting the processed data stored in `df1` into a CSV file ('suplyer.csv'). This export ensured that the gathered information could be easily shared, analyzed, or integrated into other applications for business insights or decision-making processes.

Cost Estimation Algorithm

The `estimate_cnc_part_cost_range` function provides a robust framework for estimating the manufacturing costs associated with CNC (Computer Numerical Control) parts. This algorithm leverages several key parameters—material type, material usage, part size, and complexity—to compute both minimum and maximum cost estimates. Each parameter is meticulously evaluated using predefined industry standards and cost factors, ensuring accuracy and reliability in cost predictions.

Parameters and Cost Factors

The algorithm begins by defining material costs per unit, encapsulated within the `material_costs` dictionary. This dictionary categorizes various materials commonly used in CNC machining, such as aluminum, steel, and plastics, each assigned a specific cost reflective of market rates and material properties.

```
material_costs = {
        'Aluminum': 2.5,
        'Steel': 3.0,
        'Plastic': 1.5,
        'MS/SS': 4.0,
        'Stainless Steel': 3.25,
        'Carbon Steel': 2.5,
        'Mild Steel': 1.75,
        'MS': 2.0,
        'Cast Iron': 2.25,
        'Alloy Steel': 3.0,
        'En - 36': 5.0,
    }
```

Machine time costs are categorized based on part size—small, medium, and large—and further classified by complexity levels (simple, moderate, complex) within the `machine_time_costs` dictionary. This classification structure allows for precise calculation of the time required for CNC machining operations, influencing overall production costs accordingly.

```
machine_time_costs = {
        'Small': {'Simple': 10, 'Moderate': 15, 'Complex': 20},
        'Medium': {'Simple': 20, 'Moderate': 30, 'Complex': 40},
        'Large': {'Simple': 30, 'Moderate': 45, 'Complex': 60}
    }
```

Setup costs, inherent to the complexity of CNC part production, are stored in the `setup_costs` dictionary. These costs represent the initial setup and preparation expenses involved in machining different part complexities.

```
  setup_costs = {
        'Simple': 50,
        'Moderate': 75,
        'Complex': 100
    }
```

Cost Calculation

The algorithm validates user inputs for material type, part size, and complexity to ensure compatibility with predefined options. Any invalid input triggers a `ValueError`, maintaining robustness in data processing and computation.

```
if material not in material_costs:
        raise ValueError("Invalid material type.")
    if part_size not in machine_time_costs:
        raise ValueError("Invalid part size.")
    if complexity not in machine_time_costs[part_size]:
        raise ValueError("Invalid complexity level.")
```

Cost estimation proceeds by calculating each component necessary for CNC part production. Material costs are computed by multiplying the unit cost of the selected material (`material_costs[material]`) by the specified material usage. Machine time costs are derived from `machine_time_cost` based on the selected part size and complexity level, directly influencing production timelines and expenditures.

```
  material_cost = material_costs[material] * material_usage
    machine_time_cost = machine_time_costs[part_size][complexity]
    setup_cost = setup_costs[complexity]
```

Tooling costs, critical in CNC machining, are determined based on the complexity of the part and the estimated usage hours per tool. The algorithm accounts for both minimum and maximum tool costs per hour, adjusting calculations to reflect the variability in machining requirements.

```
tool_cost_per_hour_min = tool_costs_per_hour[complexity][0]
    tool_usage_hours_min = tool_usage_hours[complexity][0]
    tool_cost_min = tool_usage_hours_min * tool_cost_per_hour_min
    tool_cost_per_hour_max = tool_costs_per_hour[complexity][1]
    tool_usage_hours_max = tool_usage_hours[complexity][1]
```

Labor costs, a fundamental component in manufacturing operations, are standardized at a fixed hourly rate (`labor_cost_per_hour`). These costs are directly proportional to machine time, reflecting the labor-intensive nature of CNC part production.

```
    tool_cost_max = tool_usage_hours_max * tool_cost_per_hour_max
```

Overhead costs, representing supplementary expenses such as administrative overhead and facility maintenance, are calculated as a percentage of the total direct costs (material, machine time, setup, tooling, and labor). This ensures comprehensive coverage of all ancillary expenses associated with CNC machining operations.

```
overhead_cost_min = overhead_cost_percentage * (material_cost + machine_time_cost +
setup_cost + tool_cost_min + labor_cost)

overhead_cost_max = overhead_cost_percentage * (material_cost + machine_time_cost +
setup_cost + tool_cost_max + labor_cost)
```

Supplier Matching Algorithm

The `match_suppliers` function selects suppliers based on specified part specifications and cost constraints. It iterates through a DataFrame of suppliers, checking if each supplier offers the required material and quotes a price within the defined range (`min_cost` to `max_cost`). For each matching supplier, the function calculates the distance from a base location using a simulated distance calculation function. The results are compiled into a dictionary (`matched_suppliers`) containing essential details such as supplier name, location, material availability, quoted price, and distance. This approach efficiently identifies suitable suppliers, supporting decision-making in CNC part procurement based on both technical specifications and logistical considerations.

Challenges Faced During the Project

One of the primary challenges was identifying a suitable website to scrape data from, as the recommended sources in the documentation did not permit web scraping. To address this, extensive research was conducted to find alternative websites that offered the necessary data and allowed scraping. Another significant challenge was determining the appropriate formula for CNC cost calculation. This required a thorough search on the internet to gather reliable information and ensure the accuracy of the cost estimation algorithm. Both challenges were overcome through diligent research and adapting the project requirements to available resources.

Ideas for Future Improvements and Scaling

Future improvements for this project could include integrating the `Geopy` library and Google API keys to calculate the exact distance between the supplier and our base location. This would enhance the accuracy of the logistics planning and cost estimation by providing precise distance measurements. Additionally, improving data collection through more effective web scraping techniques could help optimize the product further. By sourcing comprehensive and up-to-date supplier information, the system could offer better supplier matches, enhancing the overall efficiency and effectiveness of the procurement process.