Clustering Startups

April 21, 2025

1 Successful Startups and Common Indicators Between Them

1.1 Purpose and Questions

Throughout our society today, startups have become a central force in driving innovation, job creation, and economic growth. The number of startups reaching billion-dollar "unicorn" valuations has grown rapidly in recent years. However, the factors that lead some startups to achieve such success while others fail remain widely debated. Many people point to location, industry, total funding, number of investors, or even the background of the founders. Despite the variety of explanations, no single factor has emerged as the definitive key to success.

Throughout this paper, I hope to explore the key characteristics that correlate with a startup reaching unicorn status and investigate whether those correlations imply causation or if deeper structural patterns are at play.

1.1.1 Questions to Answer

- Does the country or city in which a startup is founded significantly affect its likelihood of reaching unicorn status?
- Does the amount of total funding or number of funding rounds correlate with higher startup valuation?
- Do certain industries have a higher success rate in producing unicorns?

I will be using a dataset from Kaggle that includes information on over 1,000 startups valued at $1 billion or more. You can access the dataset with this link: https://www.kaggle.com/datasets/thedevastator/startups-valued-at-1-billion-or-more. I will focus on uncovering what makes these startups different and what lessons we can draw from their success.

1.2 Introduction to Clustering

Clustering is an **unsupervised** learning technique that finds natural groupings without labels required. Two common methods:

- **K-Means**
  1. Pick a number of clusters k.

  2. Randomly initialize k centroids.

  3. Assign each point to nearest centroid.

4. Recompute centroids → repeat until stable.

   - **Pros:** Fast, scales well, easy to implement.

   - **Cons:** Must choose k, sensitive to outliers & feature scale.

- **Agglomerative (Hierarchical)**
  1. Start with each point as its own cluster.

  2. Merge the two closest clusters (using a linkage metric).

  3. Repeat until you have the desired number of clusters.

  - **Pros:** No fixed centroids, produces a dendrogram, flexible shapes.

  - **Cons:** $O(n^2)$ or worse for large n, merging is final.

1. **Preprocess**
   - Handle nulls (e.g. fill/drop), drop irrelevant columns, check for duplicates.

2. **EDA & Visualizations**
   - With the cleaned data, plot histograms, bar charts, scatter plots to spot distributions, trends, and outliers.

3. **Scale Features**
   - Standardize numeric columns (valuation, year) so no feature dominates distance calculations.

4. **Choose *k***
   - Use the elbow method (inertia vs. k) and silhouette scores to identify the optimal number of clusters.

5. **Fit Models**
   - Apply K-Means and/or Agglomerative with the chosen k.

6. **Evaluate**
   - Compute silhouette score to gauge cluster quality.

7. **Profile & Interpret**
   - Aggregate cluster statistics (means, modes) and translate each group into real-world insights.

[6]:

```python
#first, let's load the dataset, we'll need pandas
import pandas as pd

df = pd.read_csv('unicorns.csv')
```

[7]:

```python
df.shape #rows x columns
```

```
[7]: (1199, 11)
```

1.3 Preprocessing

Before we start visualizing data, we should evaluate the dataset and determine whether the dataset needs any cleaning done.

```
[9]: print(df.isnull().sum())   #check for nulls in every column
     print(df.isnull().sum().sum())
```

```
Updated at                       0
Company                          0
Crunchbase Url                   0
Last Valuation (Billion $)       0
Date Joined                      0
Year Joined                      0
City                            18
Country                          0
Industry                         0
Investors                        0
Company Website               1199
dtype: int64
1217
```

```
[10]: df['City'] = df['City'].fillna('Unknown')

      df = df.drop(columns=['Company Website'])
      df = df.drop(columns=['Crunchbase Url'])
      df = df.drop(columns=['Updated at'])
      # Check if nulls are removed
      print(df.isnull().sum())
```

```
Company                       0
Last Valuation (Billion $)    0
Date Joined                   0
Year Joined                   0
City                          0
Country                       0
Industry                      0
Investors                     0
dtype: int64
```

```
[11]: print(df.duplicated().sum())  # Count duplicate rows
```

```
0
```

For preprocessing, I started by checking for missing values and found that the ‘City’ column had 18 nulls, while ‘Company Website’ had over 1,100 missing values. Since city data could still be useful, I replaced its nulls with ‘Unknown’ to retain those rows. On the other hand, I dropped the ‘Company Website’ column and ‘Crunchbase Url’ because neither felt pertinent to this project.

After these adjustments, I confirmed there were no remaining null values in the dataset. Finally, I checked for duplicate rows and found none, so no further cleaning was needed.

1.4 Introduce Data

I obtained the dataset of unicorn startups from Kaggle: Startups Valued at $1 Billion or More cite turn0file0 . The raw CSV contains 1,199 rows and 11 columns; after dropping the ‘Company Website’ column (100% missing) and filling nulls in ‘City’ with “Unknown”, the final dataset has the following features:

[14]:
```
df.head()
```

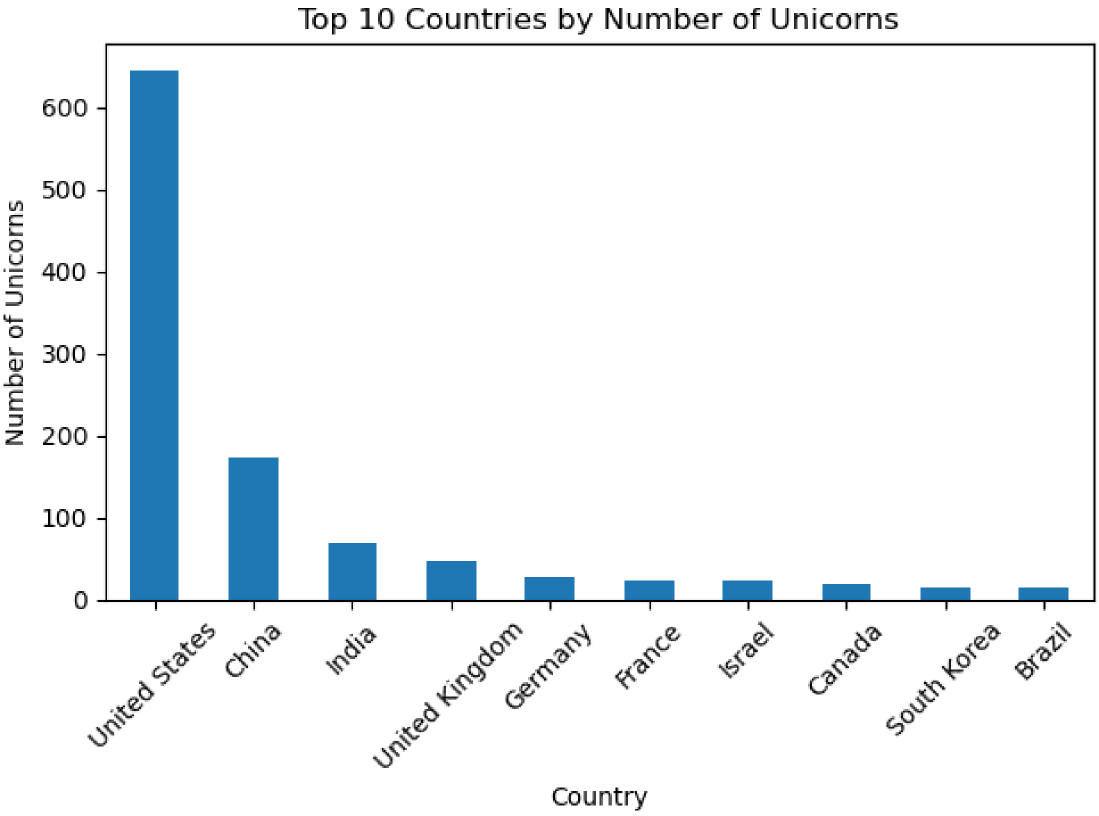
[14]:
```
      Company  Last Valuation (Billion $) Date Joined  Year Joined  \
0       Esusu                         1.0   1/27/2022         2022   
1  Fever Labs                         1.0   1/26/2022         2022   
2       Minio                         1.0   1/26/2022         2022   
3   Darwinbox                         1.0   1/25/2022         2022   
4     Pentera                         1.0   1/11/2022         2022   

          City        Country                      Industry  \
0     New York  United States                       Fintech   
1     New York  United States  Internet software & services   
2    Palo Alto  United States   Data management & analytics   
3    Hyderabad          India  Internet software & services   
4  Petah Tikva         Israel                Cybersecurity    

                                           Investors  
0  ["Next Play Ventures","Zeal Capital Partners",…  
1                        ["Accel","14W","GS Growth"]  
2  ["General Catalyst","Nexus Venture Partners","…  
3  ["Lightspeed India Partners","Sequoia Capital …  
4   ["AWZ Ventures","Blackstone","Insight Partners"]  
```

1.5 Visualizations

1.5.1 Top 10 Countries by Number of Unicorns

I created this visualization to identify which countries are leading in the number of unicorn startups. Since my dataset includes information about company locations, analyzing the distribution of unicorns by country helps reveal geographic patterns in startup growth and innovation. This can be especially useful for understanding which regions have the strongest startup ecosystems and where most high-valuation companies are emerging.

[17]:
```
import matplotlib.pyplot as plt

top_countries = df['Country'].value_counts().head(10)
top_countries.plot(kind='bar')
plt.title('Top 10 Countries by Number of Unicorns')
plt.xlabel('Country')
plt.ylabel('Number of Unicorns')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



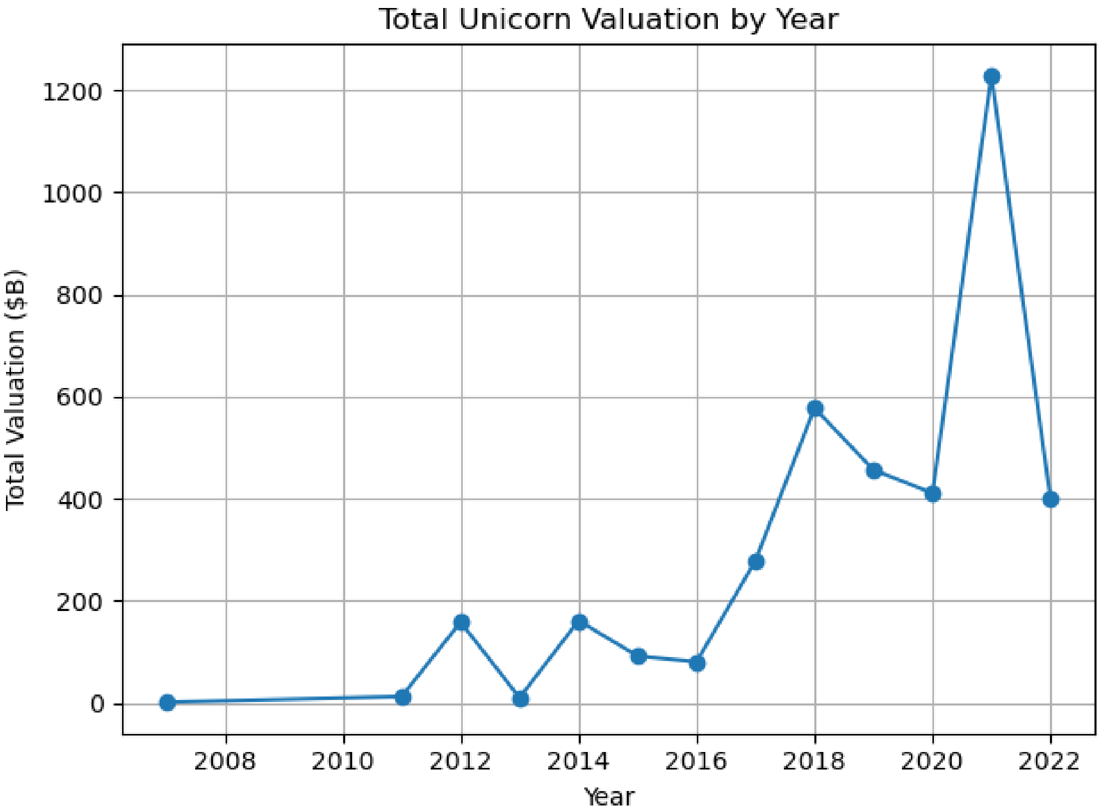
The bar chart displays the top 10 countries by the number of unicorns. The United States leads by a wide margin, with over 600 unicorns, followed by China and India. After the top three, the numbers drop significantly, with countries like the UK, Germany, and France trailing far behind. This sharp disparity highlights the dominance of the U.S. and China in the global startup scene, while also showing which other countries are notable contributors.

1.5.2 Unicorn Valuations Over Time

I wanted to understand how unicorn company valuations have evolved over time, so I analyzed the total valuation of unicorns by the year they joined the list. By converting the 'Date Joined' column into a datetime format and extracting the year, I was able to group the data and observe broader trends. This visualization helps uncover whether there's been significant growth in startup valuations over the past decade and highlights any standout years.

[20]:
```
df['Date Joined'] = pd.to_datetime(df['Date Joined'], errors='coerce')
df = df.dropna(subset=['Date Joined'])
```

```
df['Year Joined'] = df['Date Joined'].dt.year
valuation_by_year = df.groupby('Year Joined')['Last Valuation (Billion $)'].
 ↪sum()

valuation_by_year.plot(kind='line', marker='o')
plt.title('Total Unicorn Valuation by Year')
plt.xlabel('Year')
plt.ylabel('Total Valuation ($B)')
plt.grid(True)
plt.tight_layout()
plt.show()
```



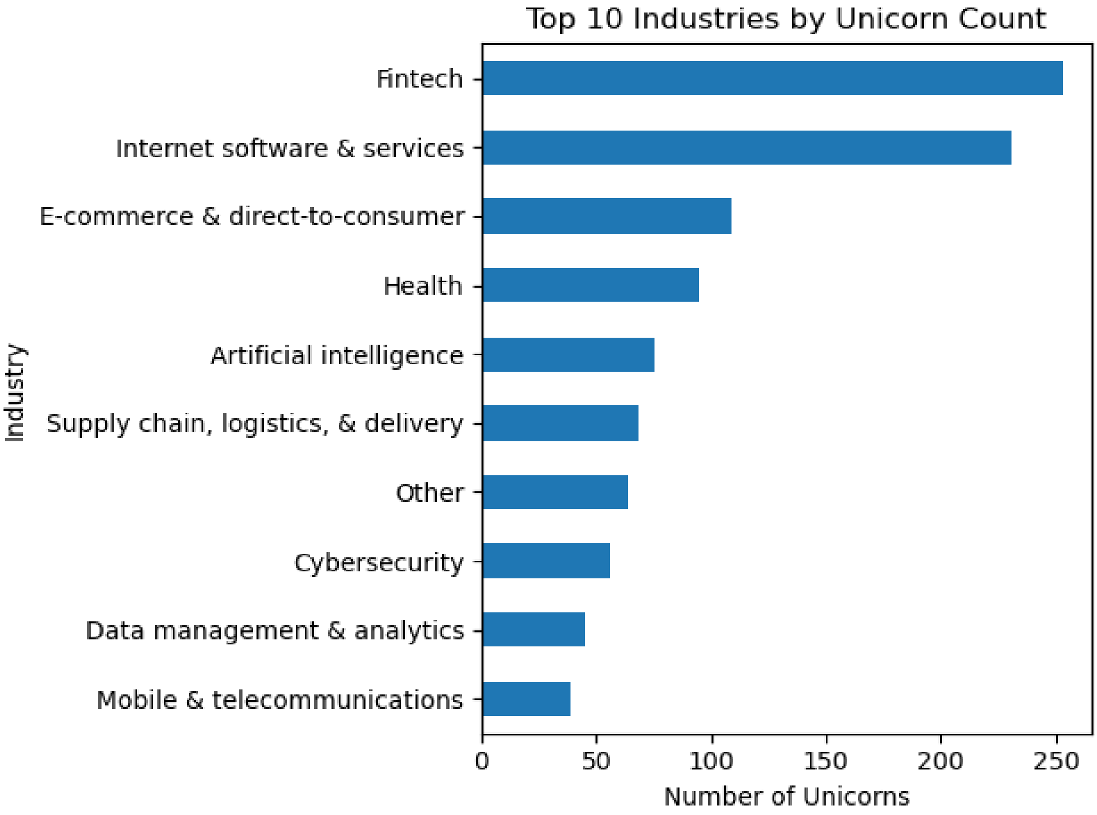
The line graph illustrates the total valuation of unicorns by year, showing a dramatic increase starting around 2017, with a massive spike in 2021. This suggests a boom period in the startup ecosystem, likely influenced by high investor activity and inflated valuations during the pandemic era. The drop after 2021 may point to market corrections or fewer new unicorns joining at extremely high valuations. Overall, the graph reveals how startup valuations have not grown linearly but have instead surged in waves.

1.5.3 Top 10 Industries with Most Unicorns

To understand where the majority of unicorn startups are concentrated, I visualized the top 10 industries with the most unicorns. Knowing the dominant industries can help identify which sectors are currently driving innovation and attracting the most investment. This analysis is also valuable for clustering later on, as industry can be a key categorical feature when grouping similar startups.

[23]:

```
top_industries = df['Industry'].value_counts().head(10)
top_industries.plot(kind='barh')
plt.title('Top 10 Industries by Unicorn Count')
plt.xlabel('Number of Unicorns')
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```



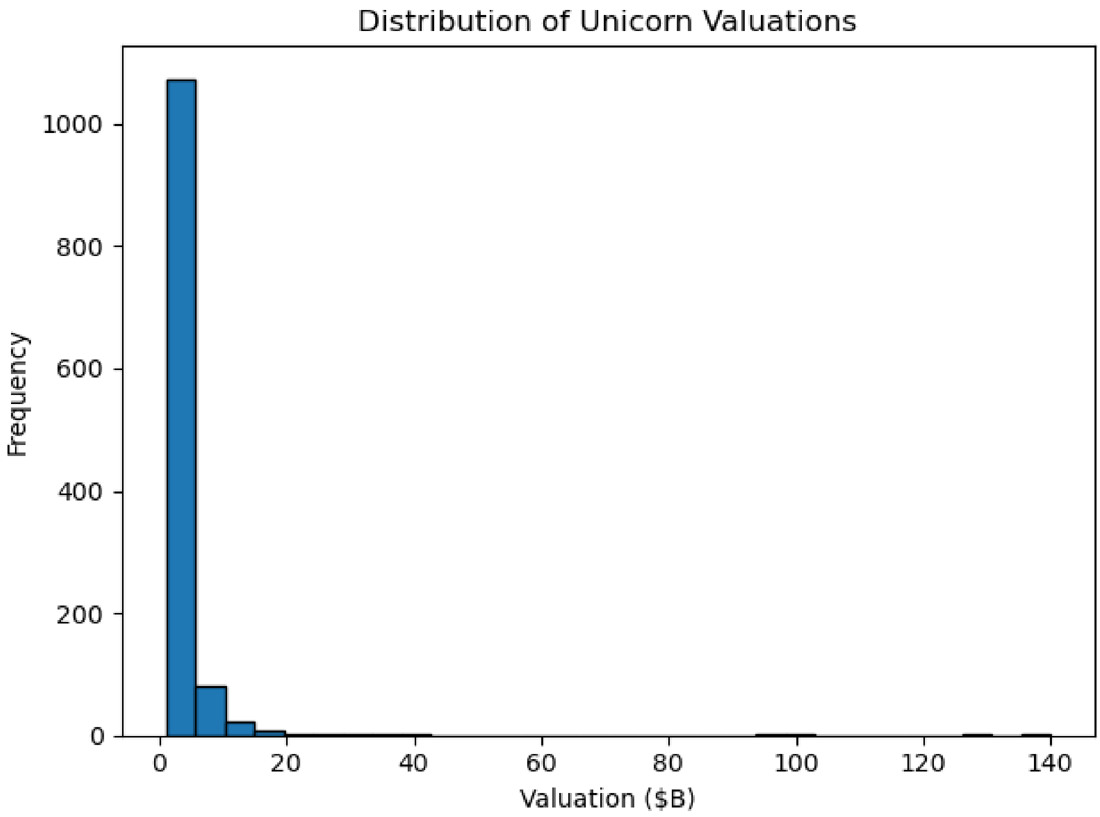
The horizontal bar chart displays the top 10 industries by number of unicorns. Fintech leads the list, followed closely by Internet software & services, showing that technology-heavy sectors dominate the unicorn space. E-commerce, health, and AI also have significant representation, pointing to strong startup activity in consumer platforms, healthcare innovation, and emerging tech. The distribution gives a clear picture of where investor interest and startup growth are most concentrated.

1.5.4 Valuation Distribution

I wanted to get a sense of how unicorn valuations are distributed across the dataset. Visualizing the distribution with a histogram helps reveal whether most unicorns are valued similarly or if there are outliers with exceptionally high valuations. This insight is important for understanding the range and concentration of startup valuations and can help inform how I treat this feature during normalization or clustering.

[26]:

```
df['Last Valuation (Billion $)'].plot(kind='hist', bins=30, edgecolor='black')
plt.title('Distribution of Unicorn Valuations')
plt.xlabel('Valuation ($B)')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



The histogram shows that the vast majority of unicorns have valuations clustered on the lower end, mostly under USD 10 billion. There's a steep drop-off as valuations increase, with only a few companies reaching extremely high valuations beyond USD 40B. This right-skewed distribution highlights that while a few unicorns are worth tens of billions, most fall closer to the minimum threshold for unicorn status, around USD 1B – USD 2B.

1.6 Introduction to Clustering

1.7 Modeling

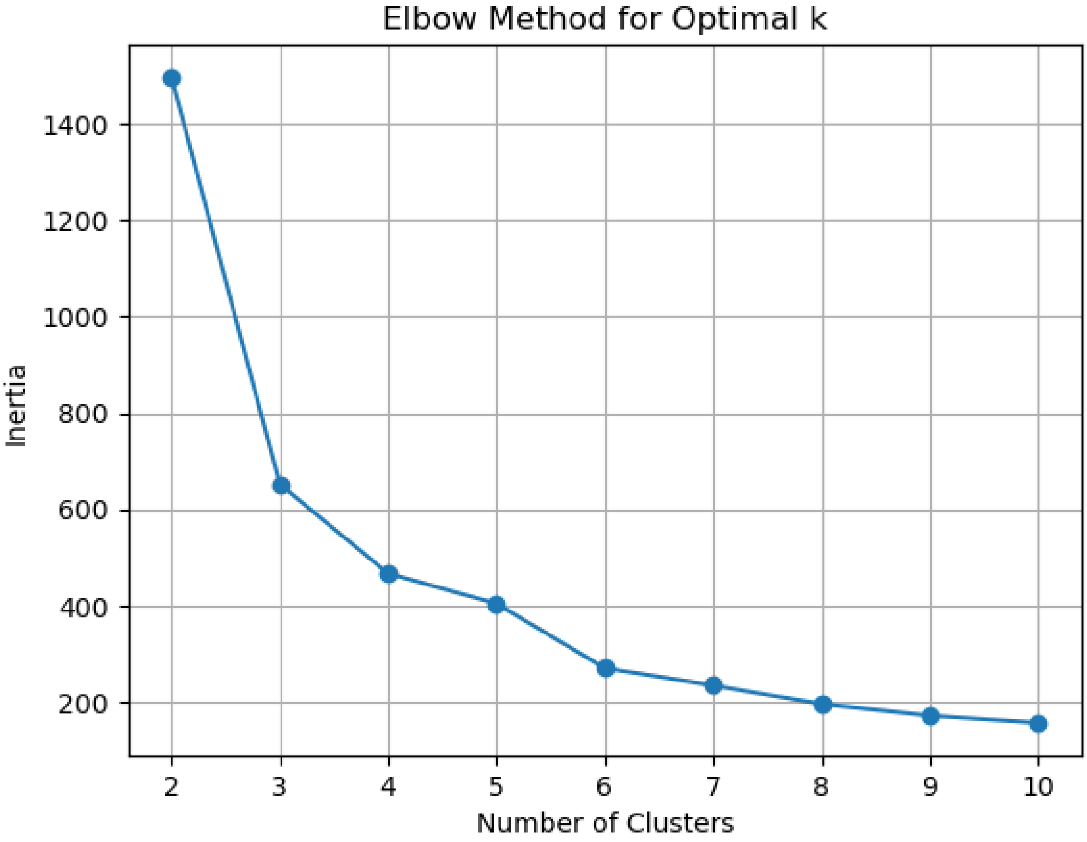
1.7.1 K-Means

To group startups by their valuation and founding year, I applied K-Means clustering. The valuation histogram revealed four natural bands (core $1–2 B, mid USD 3-10 B, high USD 10–40 B, and extreme >USD 40 B), so I first standardized "Last Valuation" and "Year Joined" with a StandardScaler to balance their influence. I then ran the elbow method, plotting inertia for k=2–10 and observed the most pronounced bend at k=4, exactly matching those four bands. Finally, I evaluated the resulting clusters using the silhouette score, which measures how well each point fits within its assigned cluster (with values closer to 1 indicating clearer separation). The code below implements these steps, selects k=4, fits the model, and reports the silhouette score.

[31]:

```python
from sklearn.preprocessing import StandardScaler

# Select numerical features for clustering
features = df[['Last Valuation (Billion $)', 'Year Joined']].copy()

# Standardize features
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.metrics import silhouette_score

# Find optimal k using Elbow Method
inertia = []
K = range(2, 11)
for k in K:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_features)
    inertia.append(kmeans.inertia_)

plt.plot(K, inertia, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.grid(True)
plt.show()

# Assume k=4 based on elbow curve
kmeans = KMeans(n_clusters=4, random_state=42)
df['KMeans Cluster'] = kmeans.fit_predict(scaled_features)

# Evaluate with Silhouette Score
score = silhouette_score(scaled_features, df['KMeans Cluster'])
```

```
print(f"Silhouette Score (K-Means): {score:.3f}")

profile_km = df.groupby('KMeans Cluster').agg({
    'Last Valuation (Billion $)': ['count', 'mean', 'min', 'max'],
    'Year Joined': ['mean'],
    'Industry':   lambda x: x.mode()[0],
    'City':       lambda x: x.value_counts().idxmax()
}).reset_index()

# Flatten the multi-level columns
profile_km.columns = [
    'Cluster', 'Count', 'Avg-Valuation', 'Min-Valuation', 'Max-Valuation',
    'Avg-Year', 'Top-Industry', 'Top-City'
]

print(profile_km)
```



```
Silhouette Score (K-Means): 0.632
   Cluster  Count  Avg-Valuation  Min-Valuation  Max-Valuation     Avg-Year  \
0        0    865       2.153087            1.0           20.0  2021.180347
```

```
1        1     75       3.901333            1.0            39.0  2014.720000
2        2    255       4.903765            1.0            40.0  2018.368627
3        3      4     115.500000           95.0           140.0  2015.250000

                       Top-Industry       Top-City
0                           Fintech  San Francisco
1  E-commerce & direct-to-consumer        Beijing
2                           Fintech        Beijing
3           Artificial intelligence       Shenzhen
```

After fitting K Means with four clusters, we achieved a silhouette score of 0.632, indicating strong separation and cohesion. The largest cluster (865 companies) averages USD 2.15 billion in valuation and joined the unicorn club around 2021; it's dominated by fintech startups in San Francisco. A much smaller group (75 companies) averages USD 3.90 billion, dates back to 2014.72, and consists mainly of e-commerce & direct-to-consumer firms in Beijing. The third cluster (255 companies) sits in between, with an average USD 4.90 billion valuation and a 2018 join year, also largely fintech in Beijing. Finally, a tiny cluster of four outliers averages an astonishing USD 115.5 billion, joined in 2015.25, and represents leading artificial intelligence ventures based in Shenzhen. Overall, K Means has cleanly isolated the extreme outliers while grouping the vast majority of unicorns into clear, interpretable segments by valuation, vintage, industry, and location.

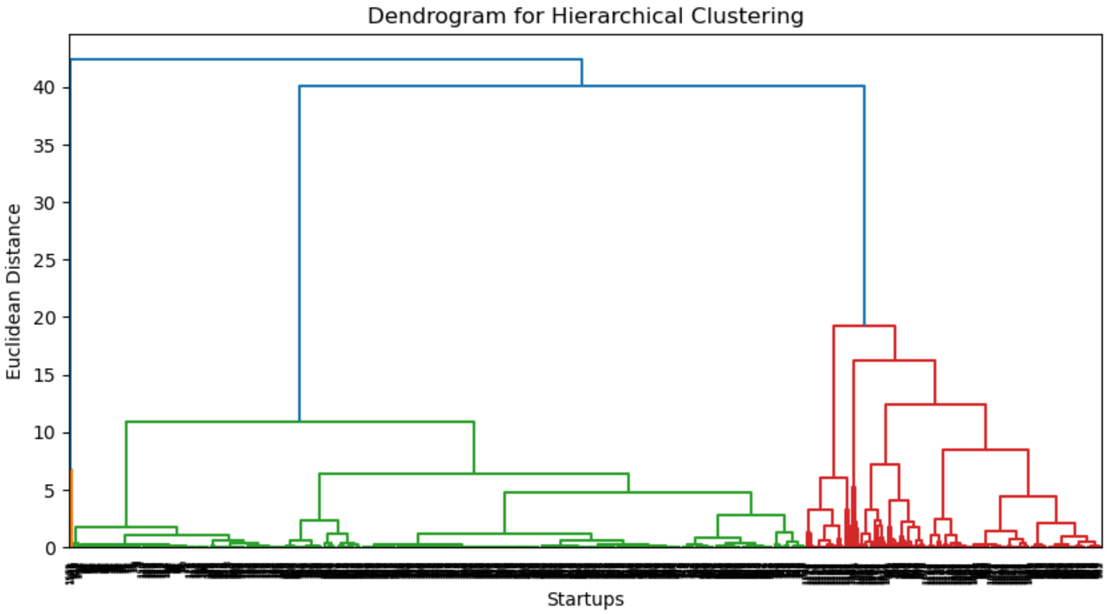
1.7.2 Agglomerative Clustering for Comparison

To compare against K-Means, I also applied Agglomerative Clustering, a hierarchical approach that builds clusters by progressively merging the closest pairs. Unlike K-Means, it doesn't require pre-defining centroids or assuming spherical clusters, making it a good alternative when the data may have irregular shapes. I used the Ward linkage method, which minimizes the variance within each cluster when merging. Although Agglomerative Clustering doesn't use inertia, I evaluated its performance using the same silhouette score metric as before for a fair comparison.

[34]:

```
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage

# Dendrogram for visualization
linked = linkage(scaled_features, method='ward')
plt.figure(figsize=(10, 5))
dendrogram(linked)
plt.title('Dendrogram for Hierarchical Clustering')
plt.xlabel('Startups')
plt.ylabel('Euclidean Distance')
plt.show()

agg = AgglomerativeClustering(n_clusters=4)
df['Agglomerative Cluster'] = agg.fit_predict(scaled_features)

agg_score = silhouette_score(scaled_features, df['Agglomerative Cluster'])
print(f"Silhouette Score (Agglomerative): {agg_score:.3f}")
```

```
# Agglomerative cluster profiling
profile_agg = df.groupby('Agglomerative Cluster').agg({
    'Last Valuation (Billion $)': ['count', 'mean', 'min', 'max'],
    'Year Joined': ['mean'],
    'Industry':   lambda x: x.mode()[0],
    'City':       lambda x: x.value_counts().idxmax()
}).reset_index()

profile_agg.columns = [
    'Cluster', 'Count', 'Avg-Valuation', 'Min-Valuation', 'Max-Valuation',
    'Avg-Year', 'Top-Industry', 'Top-City'
]

print(profile_agg)
```



```
Silhouette Score (Agglomerative): 0.623
   Cluster  Count  Avg-Valuation  Min-Valuation  Max-Valuation     Avg-Year  \
0        0    291       5.363265            1.0           40.0  2018.285223
1        1      4     115.500000           95.0          140.0  2015.250000
2        2    852       1.993357            1.0            8.5  2021.180751
3        3     52       2.815962            1.0           15.0  2014.269231

                       Top-Industry       Top-City
0                           Fintech        Beijing
1           Artificial intelligence       Shenzhen
2                           Fintech  San Francisco
3  E-commerce & direct-to-consumer        Beijing
```

Using four clusters, the agglomerative model scored 0.623 on the silhouette metric, showing that each group is more similar within itself than to the others. The algorithm first peeled off a tiny cluster of just four companies averaging 115.5 billion dollars in valuation (joined in 2015.25), all leading artificial intelligence ventures based in Shenzhen. It then formed a group of 52 startups averaging 2.82 billion (joined in 2014.27), dominated by e commerce and direct to consumer firms in Beijing. Next came a cluster of 291 companies with an average valuation of 5.36 billion (joined in 2018.29), primarily fintech startups also headquartered in Beijing. Finally, the largest cluster holds 852 unicorns averaging 1.99 billion (joined in 2021.18), mostly fintech firms in San Francisco.

This pattern tells us that ward's linkage method effectively isolates the extreme outliers first and then merges the remaining companies into groups that share similar valuation bands and founding years. The vast majority of unicorns fall into the large, moderate-valuation cluster, reflecting a recent boom in fintech out of San Francisco. Smaller clusters capture older pioneers in e commerce and a handful of ultra-high-valued AI leaders. In other words, hierarchical clustering reveals a clear hierarchy: a few dramatic success stories stand alone, while most unicorns naturally group by their size, age, industry and location.

1.7.3 Pros vs Cons of K-Means vs Agglomerative

K-Means is a fast and efficient clustering algorithm that works well when the dataset is relatively large and the clusters are roughly spherical. Its main advantage is its simplicity and speed, especially when the number of clusters is known or can be estimated using methods like the elbow curve. However, K-Means requires you to choose the number of clusters in advance, is sensitive to outliers, and assumes the clusters are similar in size and shape, which may not always match the actual data.

Agglomerative clustering does not require a predefined number of clusters and instead builds a hierarchy by merging the most similar groups step by step. This is useful for exploring the data structure when natural groupings are unclear or when cluster sizes and shapes vary. The downside is that it is more computationally expensive and does not perform as well on very large datasets.

1.8 Summary

1.8.1 Clustering Analysis

After performing clustering using both K-Means and Agglomerative models, I was able to uncover meaningful patterns among unicorn startups and gain insights into the factors that may influence their rise to billion-dollar valuations.

The clustering results show that location does play a major role. The United States, particularly cities like San Francisco and New York, dominates in both count and valuation of unicorns. This supports the idea that being based in a country with a strong startup ecosystem significantly improves the chances of success. Startups in top countries formed tighter clusters, likely benefiting from better access to funding, networks, and talent. While some unicorns emerged in other countries, the data clearly indicates that country and city are highly correlated with unicorn formation.

In terms of funding, although detailed funding round data was not included in the dataset, the valuation distribution showed that most startups cluster around the $1 to $5 billion range, with only a few extreme outliers. This suggests that while total valuation is important, reaching unicorn status does not always require excessive funding. Startups with more modest valuations still cluster closely with those that achieved similar outcomes, hinting at shared traits beyond just money raised.

As for industry, the clustering highlighted that certain sectors like FinTech, internet software, and e-commerce consistently lead in unicorn production. These industries formed strong, well-defined clusters in both models, showing that sector choice plays a critical role in startup success. The prominence of tech-related fields suggests that scalability, innovation potential, and digital infrastructure are major contributing factors.

Overall, the analysis confirmed that there is no single factor that guarantees unicorn success, but patterns do emerge. Location, industry, and valuation all show strong clustering behavior, implying that these elements are closely linked to achieving unicorn status. While clustering alone cannot prove causation, it reveals structural similarities that could guide future entrepreneurs, investors, and policymakers in fostering the next wave of successful startups.

1.8.2 Impact

This project highlights a striking pattern: the overwhelming concentration of unicorn startups in the United States, particularly in coastal cities like San Francisco and New York. The clustering analysis confirmed what many already suspect, that location significantly correlates with startup success. While this reflects the strength of these ecosystems in terms of funding, infrastructure, and networks, it also raises important concerns about equity and access. When success is so closely tied to geography, it reinforces the belief that building a billion-dollar company is only possible in a few elite cities.

As someone who works at CO LAB, UNC Charlotte's startup and innovation center, this finding feels personal. I see the potential of founders in cities like Charlotte every day. These entrepreneurs are just as capable and visionary as those in coastal hubs, but they often lack the same access to venture capital, national visibility, and support networks. The clustering results in this project illustrate how deeply rooted those barriers are. It is not simply about working hard or having a good idea, but about navigating a system that disproportionately benefits certain regions over others.

This raises important ethical questions about how the startup landscape is structured and who gets left behind. If we continue to measure success through the lens of valuation and geography, we risk overlooking innovation happening in overlooked regions. The findings in this project should encourage investors, institutions, and policymakers to reconsider how they distribute resources and who they choose to support. Rather than accepting the current trends as inevitable, we should use these insights to advocate for a more inclusive and geographically balanced startup environment.

1.9 References

GeeksforGeeks. "Agglomerative Methods in Machine Learning." GeeksforGeeks, 1 Feb. 2021, www.geeksforgeeks.org/agglomerative-methods-in-machine-learning/.

—. "K Means Clustering - Introduction." GeeksforGeeks, 30 May 2019, www.geeksforgeeks.org/k-means-clustering-introduction/.

Kaggle "Startups Valued at $1 Billion or More." Kaggle.com, 2022, www.kaggle.com/datasets/thedevastator/startups-valued-at-1-billion-or-more. Accessed 21 Apr. 2025.