Case study

plot the bar graaph between location and number of funding

Justification :

1. Data Loading:

We use pd.read_csv() to load the dataset into a pandas DataFrame, allowing easy data manipulation and analysis.

2.City Name Cleaning:

The clean_city() function standardizes city names by splitting multi-city entries, trimming spaces, and capitalizing for consistency. This helps avoid case-sensitivity issues.

3. Standardizing City Names:

df[‘City’].replace() is used to correct common variations (like "bangalore" to "Bangalore"), ensuring consistent city names.

4. Counting Funding Events:

groupby(‘City’).size() counts how many funding events occurred in each city. This method efficiently aggregates data based on city.

5. Sorting and Selecting Top Cities:

sort_values() and head(5) identify the top 5 cities with the most funding events, helping us focus on the key locations.

6. Displaying Results:

We use print() to display the top 5 cities, which helps with easy interpretation.

7. Bar Chart Visualization:

A bar chart plt.bar() visually compares the number of funding events across cities, making the results clearer and more accessible. We adjust labels for readability with plt.xticks(rotation=40).

We also use these methods?

- ‘groupby()’ and ‘size()’ allow efficient aggregation of funding events by city.
- ‘replace()’ ensures uniform city names.
- ‘sort_values()’ and ‘head(5)’ pinpoint the top locations for investment.
- A bar chart is the best way to visualize and compare categorical data like city funding counts.

In the chart we see Bangalore has highest number of funding , followed by Mumbai, New Delhi, Gurgaon, and at last is Pune.

Top 5 investor who invested maximum number of time

Justification:

1.Data Loading:

We use pd.read_csv() to load the dataset into a pandas DataFrame for easy manipulation.

2. Cleaning Investor Names:

- dropna(subset=[‘Investors Name’]) removes rows where the "Investors Name" field is empty.
- str.strip() trims extra spaces to ensure names are uniform.
- df[df[‘Investors Name’] != ‘’] further removes any entries that may still be blank after stripping.

3. Removing Undisclosed Investors:

Filtering out "Undisclosed Investors" allows us to focus on known investors, making the analysis relevant.

4. Splitting and Standardizing Investor Names:

- str.split(‘,’).explode() separates multiple investors listed together, giving each a separate row.
- str.strip() ensures clean individual entries by removing any surrounding spaces.

5. Filtering Empty Entries After Splitting:

We remove any blank entries that may have resulted from the split, ensuring accurate counts.

6. Counting and Displaying Top Investors:

- value_counts() counts each investor’s occurrences, highlighting those who have invested the most.
- head(5) extracts the top 5 investors, providing quick insight into the most active investors.

Top 5 investor who invested in different number of startups

Justification :

1. Data Loading:

Using pd.read_csv() to load the dataset provides us with a structured DataFrame for efficient analysis.

2. Removing Incomplete Data:

dropna(subset=[‘Investors Name’, ‘Startup Name’]) removes rows where key information, like the investor or startup name, is missing, ensuring only relevant data is analyzed.

3. Standardizing Startup Names:

We use replace() to consolidate common startup name variations (e.g., "Ola Cabs" and "Olacabs" to "Ola") for consistent analysis.

4. Filtering Out Irrelevant Investor Data:

We remove rows where the investor is listed as "Undisclosed Investors," as these do not contribute useful information to identify active investors.

5. Separating Multiple Investors:

- str.split(‘,’) splits multiple investors listed together into individual names.
- explode() then creates separate rows for each investor, allowing us to count individual investments accurately.
- str.strip() ensures clean, standardized investor names.

6. Removing Blank Investor Entries:

After exploding, df[df['Investors List'] != ''] removes any blank entries to maintain data quality.

7. Counting Unique Startups for Each Investor:

- drop_duplicates() ensures each investor-startup pair is unique, avoiding double-counting.
- groupby(‘Investors List’) followed by ‘.size()’ counts unique startups per investor, helping identify investors with broad startup portfolios.

8. Identifying Top Investors:

Sorting by "Unique Startups" and selecting the top 5 gives us the most active investors by unique startup investments.

Top 5 invester who invested in different startups and their investment type is "Crowd Funding" or "Seed Funding"

Justification :

1. Data Loading:

pd.read_csv() is used to load the data into a DataFrame for easy data manipulation.

2.Correcting Startup Names:

- replace() is used to standardize startup names by unifying variations (e.g., "OlaCabs" to "Ola") for consistency.

3. Standardizing Investment Types:

- replace() is applied to correct investment type names (e.g., "SeedFunding" to "Seed Funding") for accurate grouping.

4. Filtering for Early-Stage Investments:

- Filtering for "Seed Funding" and "Crowd Funding" isolates early-stage investments, focusing on initial funding activity.

5. Removing Missing Investor Names:

-dropna(subset=['Investors Name']) removes rows without investor names, ensuring only relevant data is analyzed.

6. Splitting and Expanding Investor Names:

- str.split(‘,’).explode() splits and expands multiple investors into separate rows, allowing for individual tracking.

7.Trimming Extra Spaces in Investor Names:

- str.strip() removes extra whitespace, ensuring consistency in investor names.

8. Removing Blank Entries After Splitting:

- Rows with blank investor names are removed to ensure only valid names are counted.

9. Dropping Duplicate Investor-Startup Pairs:

- drop_duplicates() ensures each unique investor-startup pairing is counted only once, focusing on unique contributions.

10. Counting Unique Startups per Investor:

- value_counts() counts unique startups for each investor, showing their level of engagement in early-stage funding.

11. Selecting Top 5 Investors by Unique Startups:

- head(5) retrieves the top 5 investors by unique startup count, highlighting the most active early-stage investors.

Top 5 invester who invested in different startups and their investment type is "Private Equity"

The justification for this is the same as above only difference is as, we know the business is in operational mode, so we just change our focus from investment type like crowd funding and seed funding to Private Equity.