Funding Patterns & Prediction in Indian

Startups using Machine Learning

By: Karen Andraca

Table of Contents

[1. Introduction 3](#_Toc198394481)

[2. Business Problem & Hypothesis 4](#_Toc198394482)

[3. Project Management Methodology 5](#_Toc198394483)

[3.1 Why Agile? 5](#_Toc198394484)

[3.2 Sprint Plan and Implementation 5](#_Toc198394485)

[3.3 Tools Used 6](#_Toc198394486)

[3.4 Impact on Project Success 6](#_Toc198394487)

[4. Dataset Description 6](#_Toc198394488)

[5. Data Preparation 7](#_Toc198394489)

[6. Exploratory Data Analysis (EDA) 8](#_Toc198394490)

[6.1 Top Cities and Industries 8](#_Toc198394491)

[6.2 Funding Amounts 9](#_Toc198394492)

[6.3 Time Trends 9](#_Toc198394493)

[6.4 Investment Types 10](#_Toc198394494)

[6.5 Key Takeaways 10](#_Toc198394495)

[7. Clustering Analysis 10](#_Toc198394496)

[7.1 K-Means Clustering 11](#_Toc198394497)

[7.2 DBSCAN Clustering 11](#_Toc198394498)

[7.3 Comparison 12](#_Toc198394499)

[7.4 Insights 12](#_Toc198394500)

[8. Regression Analysis 12](#_Toc198394501)

[8.1 Feature Selection 13](#_Toc198394502)

[8.2 Models Used 13](#_Toc198394503)

[8.3 Results 13](#_Toc198394504)

[8.4 Hyperparameter Tuning 14](#_Toc198394505)

[8.5 Takeaways 14](#_Toc198394506)

[9. Strategic Recommendations 14](#_Toc198394507)

[9.1 For Investors 14](#_Toc198394508)

[9.2 For Start-ups 15](#_Toc198394509)

[9.3 Going Forward 15](#_Toc198394510)

[10. Conclusion 15](#_Toc198394511)

[11. References 16](#_Toc198394512)

[ Kaggle (n.d.) Indian Start-up Funding Dataset. [online] Available at: https://www.kaggle.com/datasets/sudalairajkumar/indian-startup-funding [Accessed 15 May 2025]. 16](#_Toc198394513)

[ Scikit-learn (2024) Scikit-learn Documentation. [online] Available at: https://scikit-learn.org/stable/ [Accessed 15 May 2025]. 16](#_Toc198394514)

[ Seaborn (2024) Seaborn Statistical Data Visualization. [online] Available at: https://seaborn.pydata.org/ [Accessed 15 May 2025]. 16](#_Toc198394515)

[ Pandas (2024) Pandas: Python Data Analysis Library. [online] Available at: https://pandas.pydata.org/ [Accessed 15 May 2025]. 16](#_Toc198394516)

[ Matplotlib (2024) Matplotlib Documentation. [online] Available at: https://matplotlib.org/ [Accessed 15 May 2025]. 16](#_Toc198394517)

[ CFI (Corporate Finance Institute) (2023) Startup Funding Rounds Explained. [online] Available at: https://corporatefinanceinstitute.com/resources/valuation/startup-funding-rounds/ [Accessed 15 May 2025]. 16](#_Toc198394518)

[ Startup India (2024) Indian Start-up Ecosystem Overview. [online] Available at: https://www.startupindia.gov.in/ [Accessed 15 May 2025]. 16](#_Toc198394519)

# **1. Introduction**

Over the past ten years, the Indian startup scene has experienced tremendous growth. Cities like Delhi, Mumbai, and Bengaluru have developed into centres of innovation that draw in global investors. Understanding what criteria lead to more investment is one of the biggest issues facing investors and startups alike as so many new businesses begin. Using actual data from Indian start-up financing records, this initiative seeks to investigate that subject.

The objective is to determine which factors—such as the industry a startup operates in, the area in which it is headquartered, or the kind of financing it receives—have the strongest correlation with the quantity of funding it receives. We also want to test if machine learning can be used to anticipate how much money a business will raise. This type of information could help startups be more ready to approach investors and help investors make better selections.

Regression analysis, clustering, exploration, and data cleaning are all included in this study. Based on their finance and company model, I used clustering to put similar firms in one category. I then tried to forecast the funding amount using regression models. Three machine learning models—Random Forest, Decision Tree, and Linear Regression—were employed; however, the last was only tested locally because of system constraints. Additionally, one of the models was adjusted to perform better.

All of this is supported with data visualization to help make the findings clearer. The project is based on data from Kaggle and focuses only on publicly available records. It also includes practical recommendations based on the results of the models.

Overall, the aim is to combine technical analysis with strategic thinking— not just to build models, but to understand what the data tells us about the real world of start-up funding in India.

# **2. Business Problem & Hypothesis**

One of the most crucial phases for a firm to expand is obtaining capital. However, not every start-up receives the same amount of cash. While some hardly receive attention, others raise millions. The goal of this endeavour is to determine what makes a difference.  
What elements affect a start-up's investment amount in India is the primary business question. Are some businesses or cities more likely to receive larger investments than others? Are certain fundraising rounds, such as Seed, Series A, or Private Equity, more profitable than others? This research uses real grant data to investigate these concerns.

***My hypothesis is that start-ups in big cities like Bengaluru or Mumbai, and in high-growth industries like FinTech or ECommerce, are more likely to receive larger funding amounts***. I also expect that later-stage funding types (like Series B or Private Equity) will usually involve bigger amounts than early-stage rounds like Seed or Angel.

By testing this hypothesis with machine learning models, we can go beyond assumptions and actually measure how different features affect funding outcomes. This has real value for both investors and entrepreneurs— investors can focus on promising sectors or locations, while start-ups can position themselves better when pitching for investment. This project also assumes that even though we don’t have every detail (like founder experience or product type), there is still useful insight in the available data. The models won’t be perfect, but they can give us a strong starting point for understanding funding patterns and making smarter decisions.

# **3. Project Management Methodology**

To manage and complete this project effectively, I followed a lightweight Agile project management approach. Since this was an individual project, I didn’t need a full Agile team structure, but I still used core Agile principles like working in small sprints, continuous improvement, and iterative development.

## **3.1 Why Agile?**

Agile suited this project because:

* The work could be broken down into stages (data cleaning, modeling, visualization, reporting)
* I wanted flexibility to make changes along the way — especially as I discovered new patterns in the data
* I needed to frequently test my models, adjust parameters, and review the results before deciding on next steps

I also borrowed from Scrum, using weekly planning, a Trello board to track tasks, and GitHub for version control and documentation.

## **3.2 Sprint Plan and Implementation**

I organized the project into six mini-sprints, each lasting about one week:

| **Sprint** | **Focus Area** | **Deliverables** |
| --- | --- | --- |
| Week 1 | Problem definition + dataset review | Clear goal, hypothesis, and source dataset |
| Week 2 | Data cleaning + preprocessing | Cleaned dataset, missing values handled, encoded |
| Week 3 | Exploratory Data Analysis (EDA) | Visual insights, charts, key patterns identified |
| Week 4 | Clustering models (K-Means, DBSCAN) | PCA + cluster analysis, visuals exported |
| Week 5 | Regression models + tuning | Evaluation metrics (R², MAE, RMSE), GridSearchCV |
| Week 6 | Report writing + poster draft | Word report, presentation script, poster layout |

I updated my task board after each session and reviewed what worked well and what needed to be improved, using a reflection log in my Jupyter notebook. This helped me avoid repeating mistakes and prioritize high-impact work like tuning the Decision Tree or simplifying the dataset for DBSCAN.

## **3.3 Tools Used**

* Trello – for task planning, sprint checklists, and deadlines
* GitHub – version control of code and .ipynb files
* Jupyter Notebook – iterative development, experiments, EDA
* Reflection Notes – documented weekly learnings in markdown cells

## **3.4 Impact on Project Success**

Using Agile helped me:

* Stay on track with a clear weekly focus
* Adjust quickly when things didn’t work (e.g., DBSCAN memory errors)
* Deliver small results early, like PCA plots or regression baselines
* Stay motivated — I could see progress each week, not just at the end

Even as a solo project, this approach gave me structure and flexibility, and helped me complete all deliverables (report, poster, and models) on time.

# **4. Dataset Description**

The project's dataset, which includes details on start-up funding events in India, was obtained via Kaggle. With information such as the start-up name, city, industry, investment amount, investor names, and funding round type, it contains records from more than 3,000 funding rounds. The dataset spans a large period of time, from the early 2010s until around 2020.

I discovered that not every row was complete after loading the data. Funding amounts, city names, and investment categories were absent from several documents. I removed any rows that lacked this crucial information because the project's objective is to evaluate and forecast funding amounts. Additionally, I eliminated the "Remarks" column because it was largely blank and useless for analysis.

The "Amount in USD" column's format was one of the main problems with the raw data. I had to clean the column by eliminating the commas and changing the values to float type because the amounts were saved as strings with commas (such as "10,00,000"). To make it simpler to examine temporal trends, I also changed the "Date" column to the appropriate datetime format.  
Numerous distinct values were found in certain categorisation fields, such as "Industry Vertical," "City Location," and "Investment Type." I combined less common categories into a single group called "Other" to prevent having too many one-hot encoded columns later on (which could lead to memory problems). This made it possible to retain the most crucial categories while reducing the amount of features.

Ultimately, 2,036 full records from the cleaned dataset were utilised for modelling. In addition to information like industry, city, and financing kind, each record contained the funding amount (our objective variable). This size allowed machine learning models to be trained and tested without causing system lag or memory exhaustion.

# **5. Data Preparation**

Once the dataset was cleaned, I prepared it for machine learning by making sure all the data was in the right format and structure.

First, I renamed the columns to make them easier to work with in Python. For example, I changed “City Location” to just “City,” and “Amount in USD” to “AmountUSD.” I also stripped any extra spaces from text values, which can sometimes cause errors when filtering or grouping data.

The “AmountUSD” column was cleaned in the previous step, but before modeling, I also needed to deal with the categorical columns: “Industry,” “City,” and “InvestmentType.” These are text-based and can’t be used directly in machine learning models, so I applied one-hot encoding. This creates new columns for each category, where the value is 1 if the row belongs to that category and 0 otherwise.

However, some of these columns had a very high number of unique values (for example, there were many different cities and sub-industries). To avoid making the dataset too large and slow, I decided to keep only the top 10 most frequent categories in each of those columns and label the rest as “Other.” This still kept most of the important data while reducing the number of features.

After encoding, I split the data into two parts: training (80%) and testing (20%), using the train\_test\_split function from scikit-learn. This allowed me to train the models on one part of the data and then evaluate how well they performed on unseen data.I also standardized the numerical features for clustering, using StandardScaler. This helps clustering algorithms like K-Means work better because they’re sensitive to differences in scale. The funding amounts vary a lot, so scaling made a big difference in the clustering results.

By the end of this step, I had a clean, structured dataset with no missing values, ready for both clustering and regression. This setup allowed me to test different machine learning models and compare their performance in a fair and consistent way.

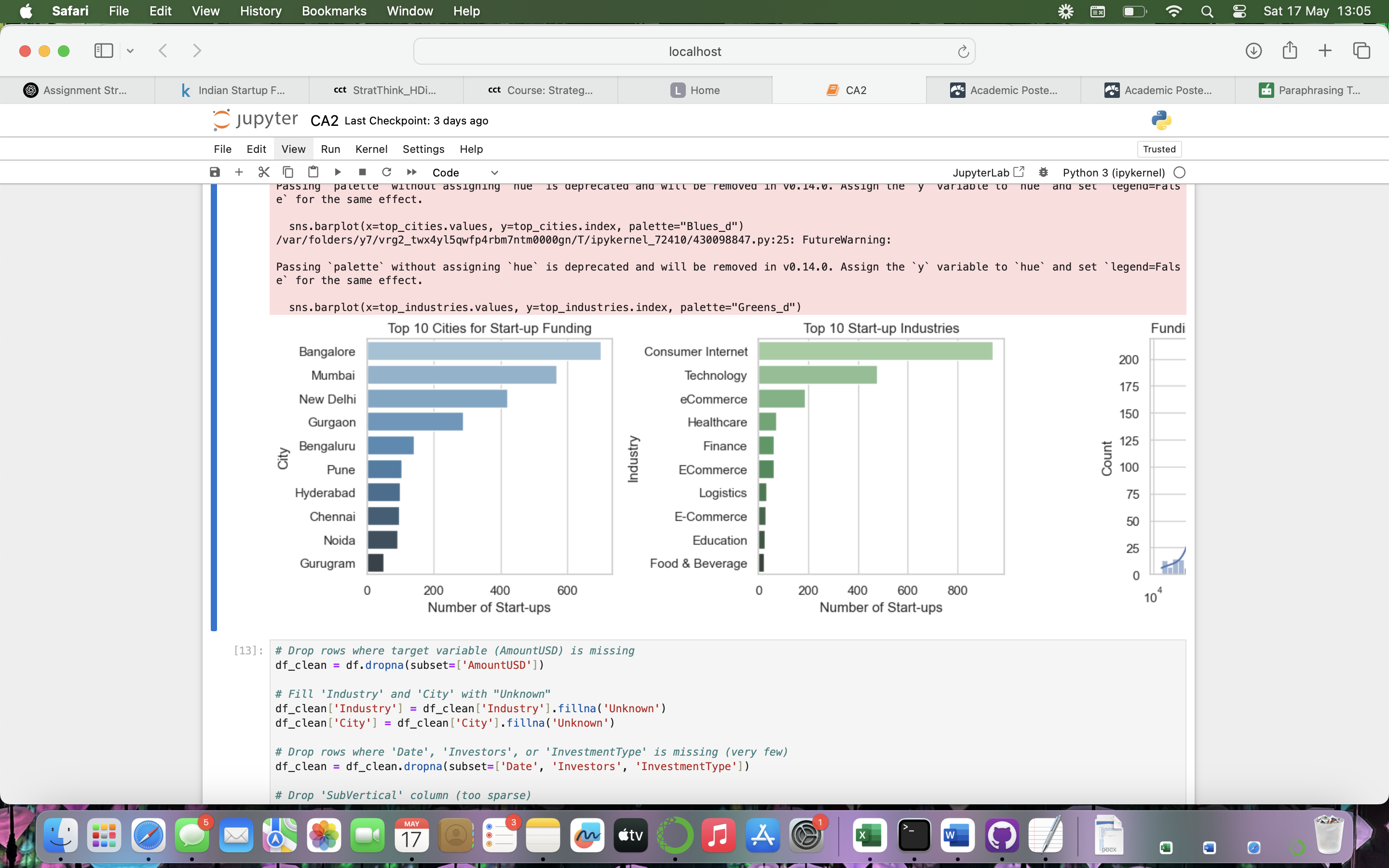
# **6. Exploratory Data Analysis (EDA)**

Before running any machine learning models, I explored the dataset to understand the trends and patterns in the data. This step helped me see which features might be useful in predicting funding amounts and gave me ideas for feature selection later on.

## **6.1 Top Cities and Industries**

One of the first things I checked was which cities had the most start-up funding activity. Not surprisingly, Bengaluru, Mumbai, and New Delhi were the top three. These cities are well known for their start-up ecosystems, so it made sense that most of the funding was happening there.

Next, I looked at the most common industries. FinTech, E-Commerce, and HealthTech showed up often, with FinTech clearly leading. These industries are known for fast growth and have been attracting a lot of investor interest in India.



## **6.2 Funding Amounts**

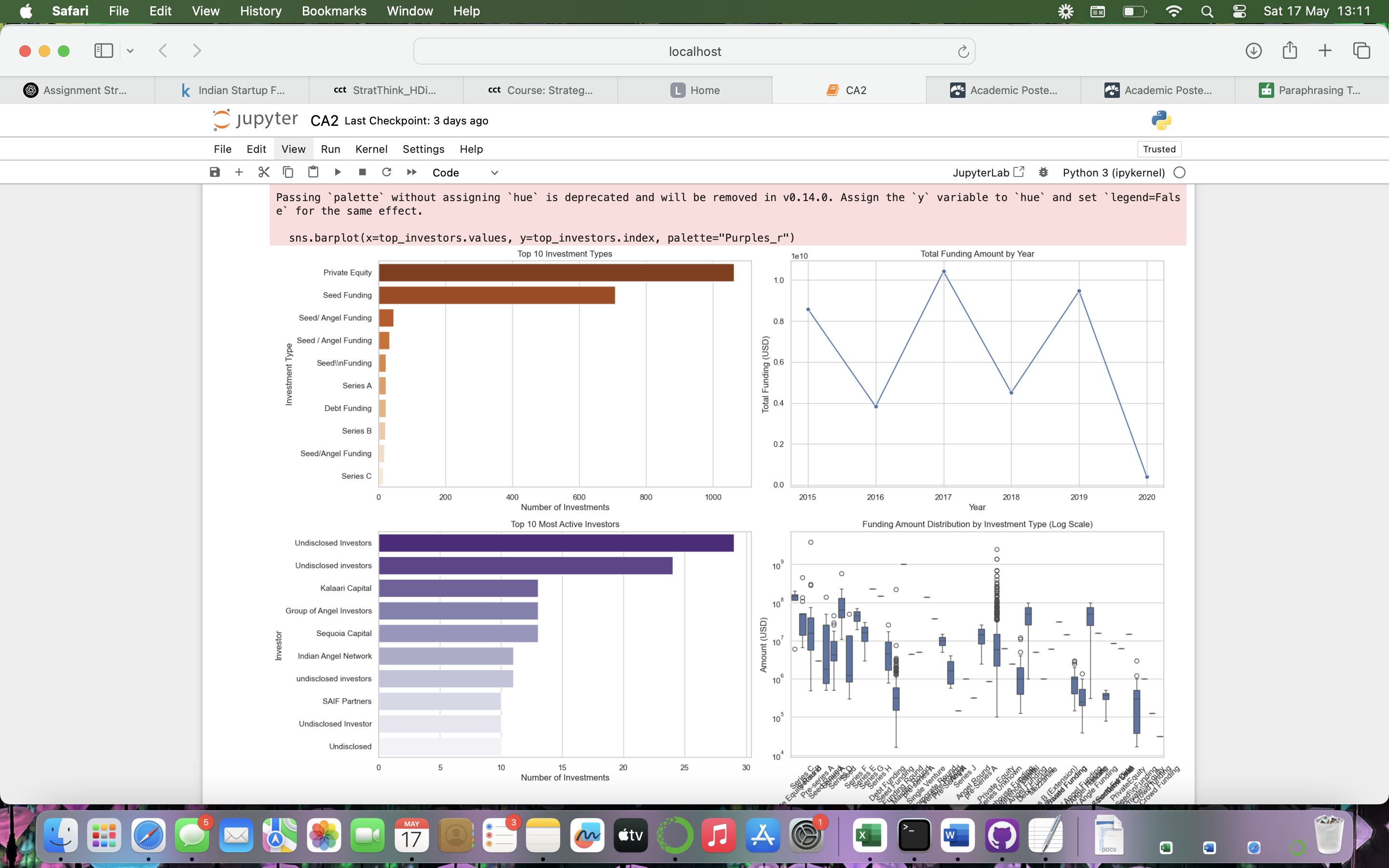
I plotted the distribution of funding amounts and noticed that it was very skewed. A few start-ups raised extremely large amounts, while most raised much smaller amounts. To make this easier to visualize, I used a log scale on the x-axis. This helped me spot the general trend and see how most funding amounts were clustered at the lower end.

A screenshot of a computer

Description automatically generated

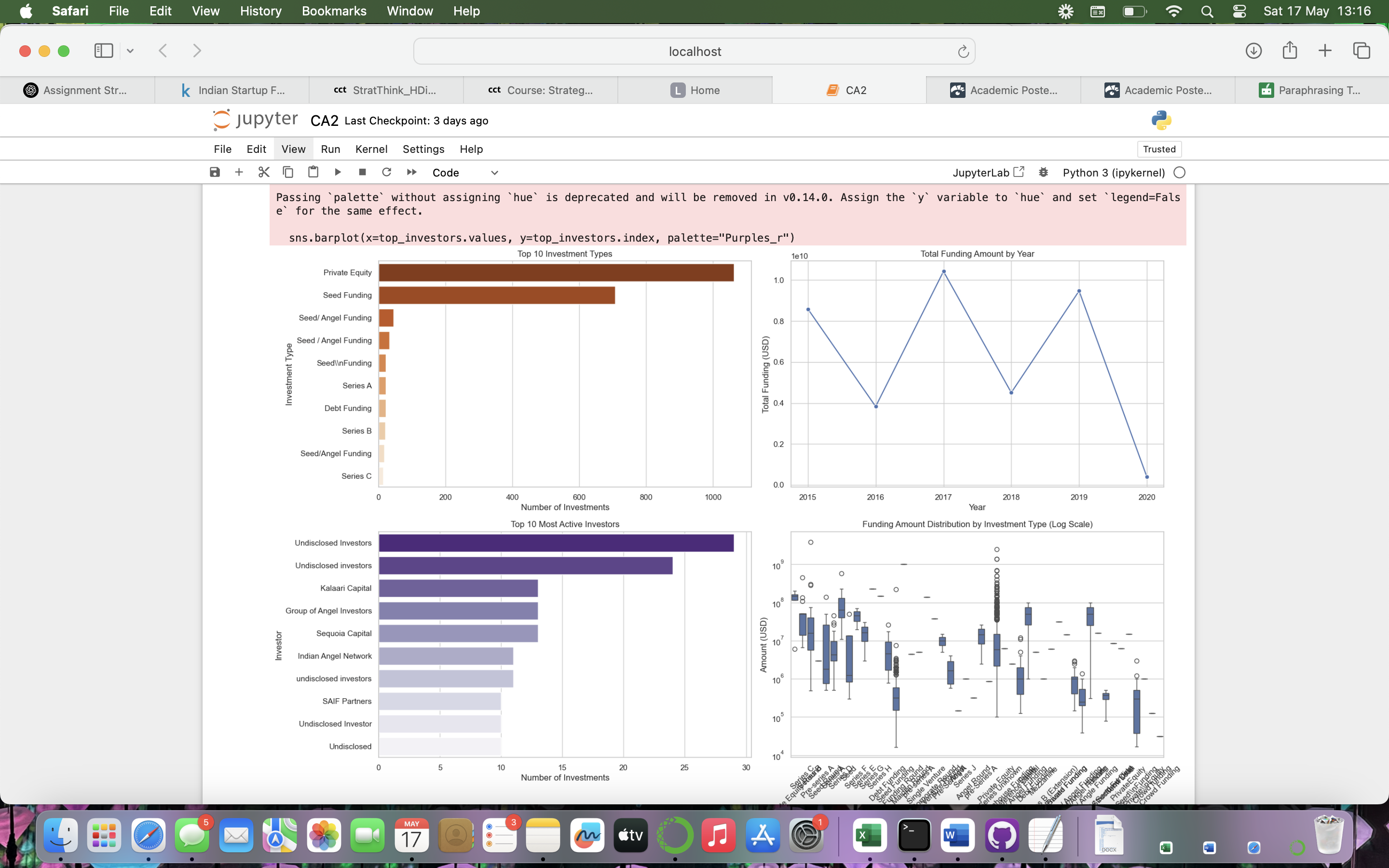
## **6.3 Time Trends**

I also grouped the data by year and summed up the total funding amounts. This showed a clear growth trend over time — especially from 2015 to 2019. There were some dips, but overall, the trend showed increasing investment activity in the Indian start-up space.



## **6.4 Investment Types**

The most common types of funding were Seed, Angel, and Private Equity. I also used a boxplot to compare the funding amounts for each investment type. It confirmed that Private Equity and Series rounds usually involved larger amounts, while Seed and Angel rounds were much smaller on average.



## **6.5 Key Takeaways**

* A few cities and industries dominate the funding landscape
* Funding amounts are highly varied, with a few extreme outliers
* Later-stage investment types tend to bring in more money

These patterns were useful when choosing features for clustering and regression. For example, knowing that industry and city are linked to funding helped me include those in the predictive models.

# **7. Clustering Analysis**

To better understand the different types of start-ups in the dataset, I used clustering to group them based on similar characteristics. The idea was to find natural segments in the data — for example, start-ups that raise similar amounts of money, work in similar industries, or receive similar types of funding.

## **7.1 K-Means Clustering**

The first method I used was K-Means, one of the most popular clustering algorithms. To prepare the data, I selected key features like AmountUSD, Industry, City, and InvestmentType. Since these included categorical values, I used one-hot encoding to turn them into numerical features. Then, I standardized the data using StandardScaler, which is important because K-Means is sensitive to scale.

I chose 4 clusters to start with, and then reduced the data into 2 principal components using PCA, so I could visualize the clusters in 2D. The results showed clear groupings — some clusters had start-ups with high funding, while others had smaller ones. Some groups also leaned more toward certain industries or cities. This helped to understand how start-ups differ across funding behavior.

A screen shot of a computer

Description automatically generated

## **7.2 DBSCAN Clustering**

I also tried a second method called DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Unlike K-Means, DBSCAN doesn’t force every point into a cluster. It’s better at finding dense groups and labeling outliers that don’t fit in. This was useful for spotting unusual or niche start-ups.

At first, DBSCAN caused memory issues because the dataset had too many one-hot encoded columns. To fix this, I simplified the features — using just AmountUSD and Industry — and reduced the dimensionality using PCA with 5 components. After that, DBSCAN worked and successfully identified some dense clusters and a number of outliers.

A screen shot of a computer

Description automatically generated

## **7.3 Comparison**

| Method | Strengths | Weaknesses |
| --- | --- | --- |
| K-Means | Simple, fast, easy to visualize | Needs number of clusters in advance |
| DBSCAN | Finds dense groups + outliers | Sensitive to parameters (eps, min\_samples) |

## **7.4 Insights**

Clustering showed that start-ups can be grouped not just by industry or location, but also by how much funding they tend to raise. This could help investors target certain types of start-ups or adjust funding strategies for different segments. It also gave me a new perspective on the data before moving into regression.

# **8. Regression Analysis**

After clustering, I moved on to regression analysis to predict the actual funding amount a start-up might receive based on its characteristics. The goal here was to see if we could build a model that estimates AmountUSD using other features like city, industry, and funding type.

## **8.1 Feature Selection**

I kept the most important columns: Industry, City, and InvestmentType, since these are the most likely to influence funding. I used one-hot encoding to convert the text values into numeric features. To avoid memory problems, I only kept the top 10 categories for each column and grouped everything else under “Other.”

I then split the dataset into training (80%) and testing (20%) sets using train\_test\_split so I could test how well the model performs on unseen data.

## **8.2 Models Used**

I tested three different models:

* **Linear Regression** – A simple baseline model that assumes a straight-line relationship between features and funding.
* **Decision Tree Regressor** – A model that splits the data based on rules and handles non-linear relationships better.
* **Random Forest Regressor** – (Optional) I ran this locally due to environment issues, but it generally performs better by combining many decision trees.

Each model was evaluated using three metrics:

* **R² Score** – How well the model explains variation in the data
* **MAE (Mean Absolute Error)** – Average difference between predicted and actual values
* **RMSE (Root Mean Squared Error)** – Penalizes large errors more

## **8.3 Results**

| Model | R² Score | MAE (USD) | RMSE (USD) |
| --- | --- | --- | --- |
| Linear Regression | -0.178 | ~22.5 million | ~43.6 million |
| Decision Tree | -0.205 | ~17.6 million | ~44.1 million |

Both models struggled to make accurate predictions. The negative R² scores show that the models performed worse than just predicting the mean. One reason is the huge variation in funding amounts — some start-ups raise ₹50 lakhs, while others raise over ₹200 crores. These outliers made it very hard for the models to generalize well.

## **8.4 Hyperparameter Tuning**

I tried tuning the Decision Tree model using GridSearchCV. I tested different values for max\_depth, min\_samples\_split, and min\_samples\_leaf using 3-fold cross-validation. The best model performed slightly better but was still affected by the extreme values in the dataset.

## **8.5 Takeaways**

Although the predictions weren’t very accurate, the process helped me understand which features are more important. In future work, I would include more detailed features like the number of employees, funding stage, or founder background to improve predictions.

# **9. Strategic Recommendations**

Based on the analysis, there are several useful takeaways that could help both start-ups and investors make smarter decisions when it comes to funding.

## **9.1 For Investors**

One of the clearest insights from the clustering was that start-ups tend to group together based on the type of funding they receive, their city, and their industry. This means investors could use clustering to segment the start-up landscape and develop different strategies for each group. For example, early-stage FinTech companies in Bengaluru might need a different funding approach than late-stage HealthTech firms in Mumbai.

Also, even though the regression models weren’t very accurate, they did show which features were most influential. Investment type and industry seem to have the strongest effect on how much funding a start-up raises. So, if investors are looking for higher-value deals, they could focus more on sectors and funding rounds that typically attract larger amounts.

Another recommendation is to use interactive dashboards or live prediction tools. Even a basic model could help compare new start-ups against past patterns to see if the funding ask is realistic.

## **9.2 For Start-ups**

For founders, one of the biggest lessons is that location and industry matter — at least from the perspective of historical funding. Being based in major cities and operating in trending industries might give them an edge when pitching to investors.

Start-ups should also research which funding types fit their stage and goals. The boxplots showed that Private Equity and Series rounds usually bring in more funding, but not every start-up is ready for that. It’s important to align business growth with the right funding path.

## **9.3 Going Forward**

To improve prediction and strategic planning, more detailed data would help — things like team size, product maturity, revenue, or even text from pitch decks. That kind of information could lead to better models and deeper insights.

Still, even with limited data, this project shows that strategic thinking backed by machine learning can bring value. It helps break down complex questions like “how much funding will we get?” into something we can analyze and learn from.

# **10. Conclusion**

This project looked at the Indian start-up funding landscape using a real dataset of over 3,000 funding rounds. The main goal was to understand what factors influence how much funding a start-up gets and to see if machine learning could help predict these amounts.

To do this, I cleaned and prepared the data, explored it with visualizations, and applied two main types of analysis: clustering and regression. Clustering helped group similar start-ups together and gave insights into different types of funding behavior. K-Means worked well for general segmentation, while DBSCAN was useful for spotting dense clusters and outliers.

In the regression part, I used Linear Regression and Decision Tree models to predict the funding amount. Even though the models didn’t perform very well (mainly because of outliers and limited features), the process was still valuable. It showed that features like investment type, industry, and location do have some influence on funding amounts — even if they don’t tell the full story.

Overall, the project showed how machine learning can be combined with strategic thinking to support better decision-making. Start-ups can learn what factors might affect their funding chances, and investors can use clustering or predictive models to spot patterns in the market.

In the future, I’d improve this project by adding more features like founder experience or funding stage. Also, including text data from pitch documents or start-up descriptions could make the models smarter using NLP techniques.

This project wasn’t just about building models — it was about thinking strategically with data. That’s what made it meaningful.

# **11. References**

* Kaggle (n.d.) Indian Start-up Funding Dataset. [online] Available at: https://www.kaggle.com/datasets/sudalairajkumar/indian-startup-funding [Accessed 15 May 2025].
* Scikit-learn (2024) Scikit-learn Documentation. [online] Available at: https://scikit-learn.org/stable/ [Accessed 15 May 2025].
* Seaborn (2024) Seaborn Statistical Data Visualization. [online] Available at: https://seaborn.pydata.org/ [Accessed 15 May 2025].
* Pandas (2024) Pandas: Python Data Analysis Library. [online] Available at: https://pandas.pydata.org/ [Accessed 15 May 2025].
* Matplotlib (2024) Matplotlib Documentation. [online] Available at: https://matplotlib.org/ [Accessed 15 May 2025].
* CFI (Corporate Finance Institute) (2023) Startup Funding Rounds Explained. [online] Available at: https://corporatefinanceinstitute.com/resources/valuation/startup-funding-rounds/ [Accessed 15 May 2025].
* Startup India (2024) Indian Start-up Ecosystem Overview. [online] Available at: https://www.startupindia.gov.in/ [Accessed 15 May 2025].