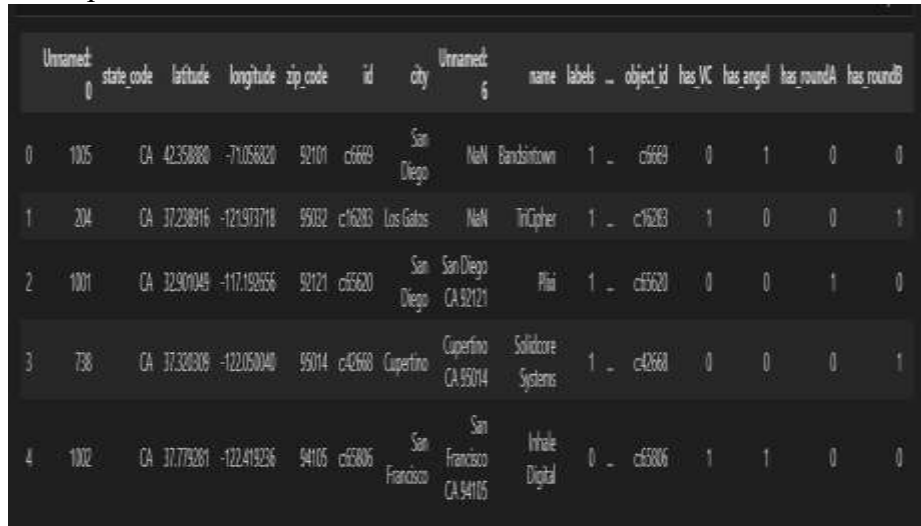


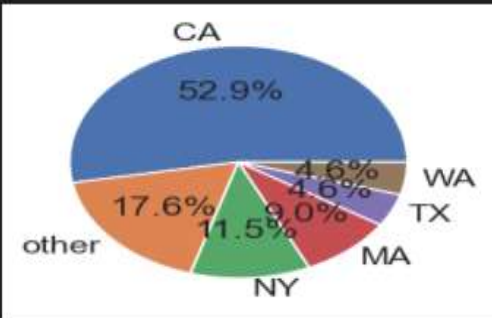
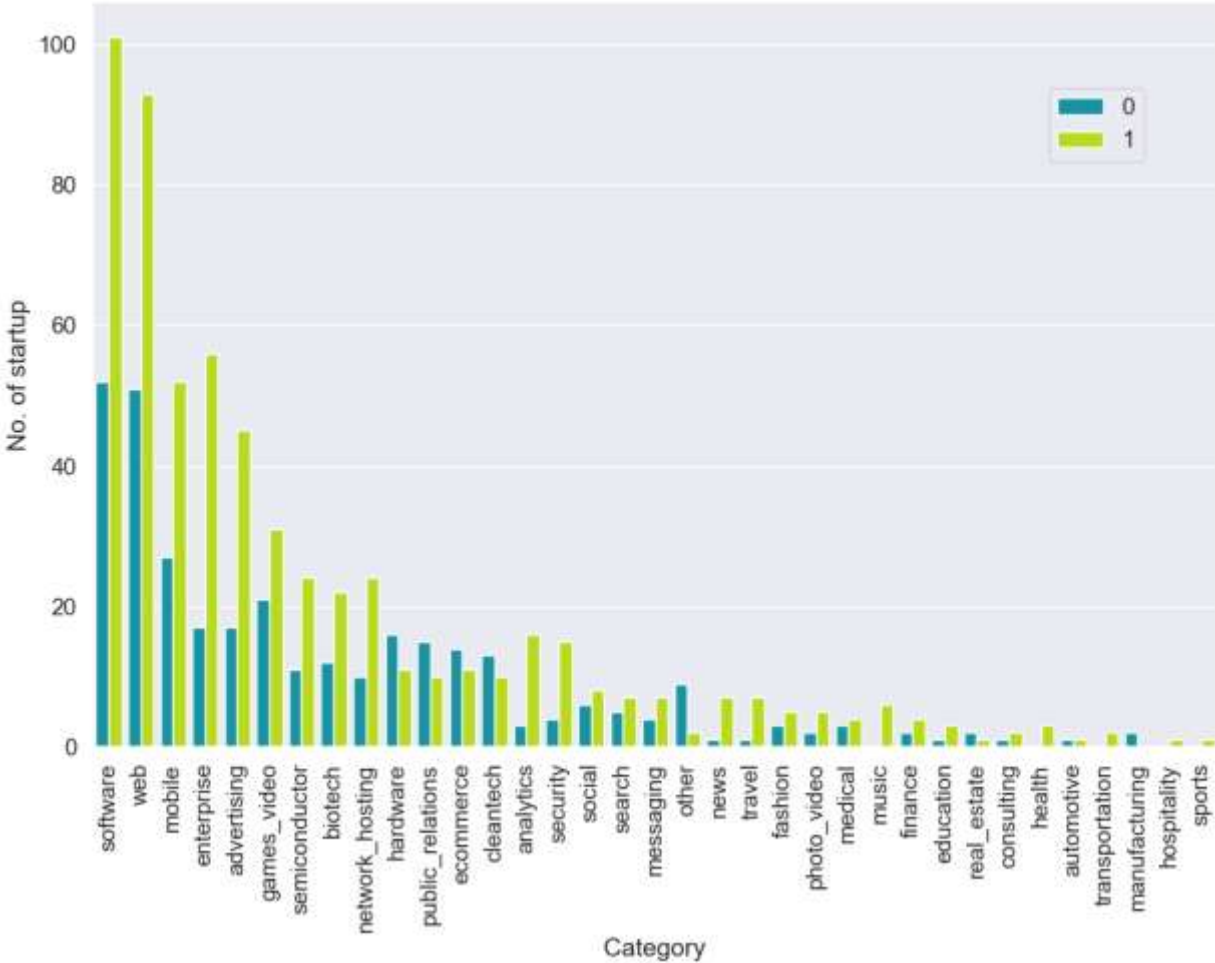
## Data Collection and Preprocessing Phase

Date	7 June 2024
Team ID	739865
Project Title	prosperity Prognosticator : Machine Learning for Startup Success Prediction
Maximum Marks	6 Marks

### Data Exploration and Preprocessing Report

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	Description
Data Overview	<p><u>Dimension:</u> 923 rows <math>\times</math> 49 columns</p> <p><u>Descriptive statistics:</u></p> 
Univariate Analysis	

	 <p>A pie chart illustrating the geographical distribution of startups. The largest segment is California (CA) at 52.9%, followed by 'other' at 17.6%, New York (NY) at 11.5%, Massachusetts (MA) at 9.0%, Texas (TX) at 4.6%, and Washington (WA) at 4.6%.</p>
Bivariate Analysis	 <p>A bar chart comparing the number of startups across 30 categories for two groups, labeled 0 (teal) and 1 (yellow-green). The y-axis represents the 'No. of startup' from 0 to 100. The x-axis lists categories: software, web, mobile, enterprise, advertising, games_video, semiconductor, biotech, network_hosting, hardware, public_relations, ecommerce, cleantech, analytics, security, social, search, messaging, other, news, travel, fashion, photo_video, medical, music, finance, education, real_estate, consulting, health, automotive, transportation, manufacturing, hospitality, and sports. Category 1 consistently shows higher counts than category 0 across most sectors, particularly in software, web, and mobile.</p>
Outliers and Anomalies	-

## Data Preprocessing Code Screenshots

### Loading Data

```
data = pd.read_csv('startup.csv')
data
```

	Unnamed: 0	state_code	latitude	longitude	zip_code	id	city	Unnamed: 6	name	labels	...	object_id	has_VC	has_angel	has_roundA	has_roundB
0	1005	CA	42.359681	-71.056820	92101	c3668	San Diego	Half	Bandtown	1	-	c3668	0	1	0	1
1	294	CA	37.238996	-121.973218	95032	c36283	Los Gatos	Half	TriCipher	1	-	c36283	1	0	0	1
2	1001	CA	32.301049	-117.189536	92121	c35620	San Diego	San Diego CA 92121	Flai	1	-	c35620	0	0	1	1
3	738	CA	37.320389	-122.030840	95014	c42668	Capetino	Capetino CA 95014	SoloCore Systems	1	-	c42668	0	0	0	1
4	1082	CA	37.770281	-122.419236	94105	c35806	San Francisco	San Francisco CA 94105	Inhalo Digital	0	-	c35806	1	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
918	352	CA	37.740594	-122.376471	94107	c21543	San Francisco	Half	Colwest	1	-	c21543	0	0	1	1
919	721	MA	42.594817	-71.295811	1803	c41747	Burlington	Burlington MA 1803	Red Point Systems	0	-	c41747	1	0	0	1
920	557	CA	37.485261	-122.015520	94089	c31548	Sunnyvale	Half	Finco Medical	0	-	c31548	0	0	0	1
921	589	CA	37.556732	-122.268778	94004	c33198	San Francisco	Half	Causata	1	-	c33198	0	0	1	1
922	482	CA	37.386778	-121.962777	95054	c26702	Santa Clara	Santa Clara CA 95054	Asompra Technologies	1	-	c26702	0	0	0	1

### Handling Missing Data

```
#filling missing value column(unnamed:6)
data['unnamed: 6'] = data.apply(lambda row: (row.city) + " " + (row.state_code) + " " + (row.zip_code) , axis = 1)

# Total Missing Values column "unnamed: 6"
totalnull = data['unnamed: 6'].isnull().sum()

print('Total Missing Values Kolom "unnamed: 6": ', totalnull)

#filling missing values of column(closed_at)
data['closed_at'] = data['closed_at'].fillna(value="31/12/2013")
totalnull = data['closed_at'].isnull().sum()

print('Total Missing Values Kolom "closed_at": ', totalnull)
```

Data Transformation	<pre>data["status"] = data.status.map({'acquired':1, 'closed':0}) data["status"].astype(int)</pre> <pre>0      1 1      1 2      1 3      1 4      0 ... 918     1 919     0 920     0 921     1 922     1 Name: status, Length: 923, dtype: int64</pre>
Feature Engineering	Attached the codes in final submission.
Save Processed Data	-