# Employee Data Analysis

In [23]:
```
#importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [20]:
```
test = pd.read_csv("C:/Users/Radhakrishnan Nair/Downloads/test.csv")
```

In [21]:
```
train = pd.read_csv("C:/Users/Radhakrishnan Nair/Downloads/train.csv")
```

In [4]:
```
train.head()
```

Out[4]:

| | employee_id | department | region | education | gender | recruitment_channel | no_of_trainings | age | previous_year_rating | length_of_service | KPIs_met >80% | awards |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 65438 | Sales & Marketing | region_7 | Master's & above | f | sourcing | 1 | 35 | 5.0 | 8 | 1 | |
| **1** | 65141 | Operations | region_22 | Bachelor's | m | other | 1 | 30 | 5.0 | 4 | 0 | |
| **2** | 7513 | Sales & Marketing | region_19 | Bachelor's | m | sourcing | 1 | 34 | 3.0 | 7 | 0 | |
| **3** | 2542 | Sales & Marketing | region_23 | Bachelor's | m | other | 2 | 39 | 1.0 | 10 | 0 | |
| **4** | 48945 | Technology | region_26 | Bachelor's | m | other | 1 | 45 | 3.0 | 2 | 0 | |

In [5]:
```
test.head()
```

Out[5]:

| | employee_id | department | region | education | gender | recruitment_channel | no_of_trainings | age | previous_year_rating | length_of_service | KPIs_met >80% | awards |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 8724 | Technology | region_26 | Bachelor's | m | sourcing | 1 | 24 | NaN | 1 | 1 | |
| **1** | 74430 | HR | region_4 | Bachelor's | f | other | 1 | 31 | 3.0 | 5 | 0 | |
| **2** | 72255 | Sales & Marketing | region_13 | Bachelor's | m | other | 1 | 31 | 1.0 | 4 | 0 | |
| **3** | 38562 | Procurement | region_2 | Bachelor's | f | other | 3 | 31 | 2.0 | 9 | 0 | |
| **4** | 64486 | Finance | region_29 | Bachelor's | m | sourcing | 1 | 30 | 4.0 | 7 | 0 | |

In [6]:
```
train.columns
```

Out[6]:
```
Index(['employee_id', 'department', 'region', 'education', 'gender',
       'recruitment_channel', 'no_of_trainings', 'age', 'previous_year_rating',
       'length_of_service', 'KPIs_met >80%', 'awards_won?',
       'avg_training_score', 'is_promoted'],
      dtype='object')
```

In [12]:
```
test.columns
```

Out[12]:
```
Index(['employee_id', 'department', 'region', 'education', 'gender',
       'recruitment_channel', 'no_of_trainings', 'age', 'previous_year_rating',
       'length_of_service', 'KPIs_met >80%', 'awards_won?',
       'avg_training_score'],
      dtype='object')
```

In [9]:
```
train.describe()
```

Out[9]:

| | employee_id | no_of_trainings | age | previous_year_rating | length_of_service | KPIs_met >80% | awards_won? | avg_training_score | is_promoted |
|---|---|---|---|---|---|---|---|---|---|
| **count** | 54808.000000 | 54808.000000 | 54808.000000 | 50684.000000 | 54808.000000 | 54808.000000 | 54808.000000 | 54808.000000 | 54808.000000 |
| **mean** | 39195.830627 | 1.253011 | 34.803915 | 3.329256 | 5.865512 | 0.351974 | 0.023172 | 63.386750 | 0.085170 |
| **std** | 22586.581449 | 0.609264 | 7.660169 | 1.259993 | 4.265094 | 0.477590 | 0.150450 | 13.371559 | 0.279137 |
| **min** | 1.000000 | 1.000000 | 20.000000 | 1.000000 | 1.000000 | 0.000000 | 0.000000 | 39.000000 | 0.000000 |
| **25%** | 19669.750000 | 1.000000 | 29.000000 | 3.000000 | 3.000000 | 0.000000 | 0.000000 | 51.000000 | 0.000000 |
| **50%** | 39225.500000 | 1.000000 | 33.000000 | 3.000000 | 5.000000 | 0.000000 | 0.000000 | 60.000000 | 0.000000 |
| **75%** | 58730.500000 | 1.000000 | 39.000000 | 4.000000 | 7.000000 | 1.000000 | 0.000000 | 76.000000 | 0.000000 |
| **max** | 78298.000000 | 10.000000 | 60.000000 | 5.000000 | 37.000000 | 1.000000 | 1.000000 | 99.000000 | 1.000000 |

In [15]:
```
train.shape
```

Out[15]:
```
(54808, 14)
```

## Missing data/values

In [22]:
```
train.isnull().sum()
```

Out[22]:
```
employee_id                0
department                 0
region                     0
education               2409
gender                     0
recruitment_channel        0
no_of_trainings            0
age                        0
previous_year_rating    4124
length_of_service          0
KPIs_met >80%              0
awards_won?                0
avg_training_score         0
is_promoted                0
dtype: int64
```

## Handling missing values

In [17]:
```
#filling the missing values for categorical terms - mode()
train['education'] = train['education'].fillna(train['education'].mode()[0])
#filling the missing values for numerical terms - mean()
train['previous_year_rating'] = train['previous_year_rating'].fillna(train['previous_year_rating'].mean())
```

In [18]:
```
train.isnull().sum()
```

Out[18]:
```
employee_id             0
department              0
region                  0
education               0
gender                  0
recruitment_channel     0
no_of_trainings         0
age                     0
previous_year_rating    0
length_of_service       0
KPIs_met >80%           0
awards_won?             0
avg_training_score      0
is_promoted             0
dtype: int64
```

## Building Baseline model

#### Calculate the arithmetic mean of the Train data set

In [7]:
```
hist_mean = np.mean(train.avg_training_score)
hist_mean
```

Out[7]:
```
63.38675010947307
```

In [8]:
```
test.age
```

Out[8]:
```
0        24
1        31
2        31
3        31
4        30
         ..
23485    24
23486    31
23487    26
23488    27
23489    40
Name: age, Length: 23490, dtype: int64
```

In [36]:
```
pred_hist_mean = pd.DataFrame()
```

In [42]:
```
# Defining the columns of the Dataframe
pred_hist_mean["Age"] = test.age
pred_hist_mean["Pred"] = 63.38
```

```
In [43]: pred_hist_mean
```

```
Out[43]:
          Age   Pred
     0     24  63.38
     1     31  63.38
     2     31  63.38
     3     31  63.38
     4     30  63.38
   ...    ...    ...
 23485     24  63.38
 23486     31  63.38
 23487     26  63.38
 23488     27  63.38
 23489     40  63.38

23490 rows × 2 columns
```

```
In [40]: def mape(y_true, y_pred):
             return np.mean(np.abs((y_true - y_pred)/y_true))*100
```

```
In [45]: pred_error = mape(test.avg_training_score, pred_hist_mean.Pred)
```

```
In [46]: pred_error
```

```
Out[46]: 18.78485287924473
```

**This baseline model has 18% error, so it's a better model.**

In [14]:
```
train['department'].value_counts()
```

Out[14]:
```
Sales & Marketing    16840
Operations           11348
Technology            7138
Procurement           7138
Analytics             5352
Finance               2536
HR                    2418
Legal                 1039
R&D                    999
Name: department, dtype: int64
```

## Percentage of people who got promoted from each department

In [16]:
```
plt.rcParams['figure.figsize'] = [10, 5]
ct = pd.crosstab(train.department,train.is_promoted,normalize='index')
ct.plot.bar(stacked=True)
plt.legend(title='is_promoted',bbox_to_anchor=(1,0.5))
```

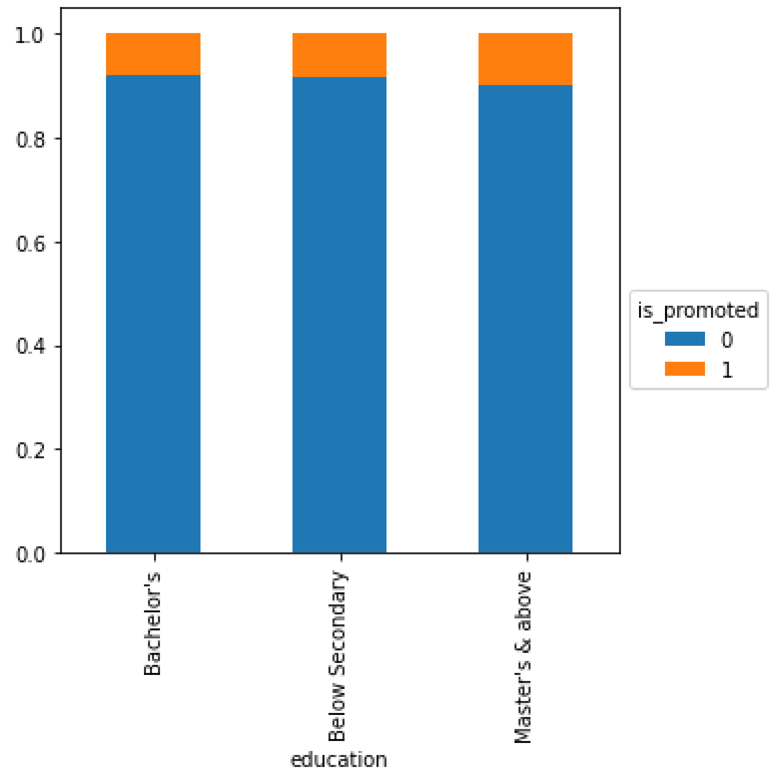
Out[16]:
```
<matplotlib.legend.Legend at 0x1da4e950790>
```

Technology department had highest percentage of employees getting promoted, Legal department has the least number. But we don't see major differences in terms of percentages.

## Distribution of promotions among people with different Educational backgrounds

```
In [17]: plt.rcParams['figure.figsize'] = [5, 5]
         edu = pd.crosstab(train.education,train.is_promoted,normalize='index')
         edu.plot.bar(stacked=True)
         plt.rcParams['figure.figsize'] = [5, 5]
         plt.legend(title='is_promoted',bbox_to_anchor=(1,0.5))
```

```
Out[17]: <matplotlib.legend.Legend at 0x1da4e950430>
```



percentages are pretty much the same aross different educational backgrounds.

## Distribution of average training score

In [19]:
```
bins = [30,40,50,60,70,80,90,100]
labels = ['30-40','40-50','50-60','60-70','70-80','80-90','90-100']
train['score_binned'] = pd.cut(train['avg_training_score'], bins=bins, labels=labels)
train['score_binned'].value_counts()
```

Out[19]:
```
50-60     16020
40-50     11996
60-70      9973
80-90      8739
70-80      7494
90-100      579
30-40         7
Name: score_binned, dtype: int64
```

While most of the employees have score in the range of 50-60, the least score bin has very faint number

### Mean score of employees with different educational background

In [20]:
```
train.groupby(["education"])['avg_training_score'].mean()
```

Out[20]:
```
education
Bachelor's          63.422046
Below Secondary     64.925466
Master's & above    64.061240
Name: avg_training_score, dtype: float64
```

Mean training score doesn't vary with education

### Filling the missing values

```
In [22]: train.isnull().any()
```

```
Out[22]: employee_id             False
         department              False
         region                  False
         education                True
         gender                  False
         recruitment_channel     False
         no_of_trainings         False
         age                     False
         previous_year_rating     True
         length_of_service       False
         KPIs_met >80%           False
         awards_won?             False
         avg_training_score      False
         is_promoted             False
         score_binned            False
         dtype: bool
```

Fill missing values of 'previous_year_rating' with mean based on 'KPIs_met >80%' and 'education' with median based on 'department'

```
In [57]: import numpy as np
         import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as plt
         %matplotlib inline
```

```
In [60]: data = pd.read_csv("C:/Users/Radhakrishnan Nair/Downloads/startup_funding.csv")
```

```
In [61]: data.columns
```

```
Out[61]: Index(['Sr No', 'Date dd/mm/yyyy', 'Startup Name', 'Industry Vertical',
                'SubVertical', 'City  Location', 'Investors Name', 'InvestmentnType',
                'Amount in USD', 'Remarks'],
               dtype='object')
```

In [62]:
```
data.head()
```

Out[62]:

| | Sr No | Date dd/mm/yyyy | Startup Name | Industry Vertical | SubVertical | City Location | Investors Name | InvestmentnType | Amount in USD | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 09/01/2020 | BYJU'S | E-Tech | E-learning | Bengaluru | Tiger Global Management | Private Equity Round | 20,00,00,000 | NaN |
| **1** | 2 | 13/01/2020 | Shuttl | Transportation | App based shuttle service | Gurgaon | Susquehanna Growth Equity | Series C | 80,48,394 | NaN |
| **2** | 3 | 09/01/2020 | Mamaearth | E-commerce | Retailer of baby and toddler products | Bengaluru | Sequoia Capital India | Series B | 1,83,58,860 | NaN |
| **3** | 4 | 02/01/2020 | https://www.wealthbucket.in/ | FinTech | Online Investment | New Delhi | Vinod Khatumal | Pre-series A | 30,00,000 | NaN |
| **4** | 5 | 02/01/2020 | Fashor | Fashion and Apparel | Embroiled Clothes For Women | Mumbai | Sprout Venture Partners | Seed Round | 18,00,000 | NaN |

In [63]:
```
data.tail(20)
```

Out[63]:

| | Sr No | Date dd/mm/yyyy | Startup Name | Industry Vertical | SubVertical | City Location | Investors Name | InvestmentnType | Amount in USD | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|
| **3024** | 3025 | 20/01/2015 | HealthyWorld.in | NaN | NaN | NaN | Group of Angel Investors | Seed Funding | 2,00,000 | NaN |
| **3025** | 3026 | 21/01/2015 | Simplilearn | NaN | NaN | NaN | Mayfield India, Kalaari Capital, Helion Ventures | Private Equity | 1,47,50,000 | NaN |
| **3026** | 3027 | 21/01/2015 | MyDreamStore | NaN | NaN | NaN | Undisclosed Investors | Seed Funding | 3,25,000 | NaN |
| **3027** | 3028 | 21/05/2015 | Knit | NaN | NaN | NaN | Rohit Jain, Amit Rambhia & Others | Seed Funding | NaN | NaN |
| **3028** | 3029 | 21/05/2015 | Villgro | NaN | NaN | NaN | Michael & Susan Dell Foundation | Seed Funding | 32,50,000 | To fund edu startups |
| **3029** | 3030 | 22/01//2015 | Corporate360 | NaN | NaN | NaN | Group of Angel Investors | Seed Funding | 2,00,000 | NaN |
| **3030** | 3031 | 22/01/2015 | Freshmonk | NaN | NaN | NaN | August Capital Partners, Michael Blakey | Seed Funding | NaN | NaN |
| **3031** | 3032 | 22/01/2015 | Englishleap.com | NaN | NaN | NaN | ANALEC | Private Equity | NaN | Majority Stake |
| **3032** | 3033 | 22/01/2015 | Antuit | NaN | NaN | NaN | Goldman Sachs, Zodius Capital | Private Equity | 5,60,00,000 | NaN |
| **3033** | 3034 | 22/01/2015 | Caratlane.com | NaN | NaN | NaN | Tiger Global | Private Equity | 3,10,00,000 | Series D |
| **3034** | 3035 | 24/01/2015 | Aasaanjobs | NaN | NaN | NaN | Inventus Capital Partners, IDG Ventures | Seed Funding | 15,00,000 | NaN |
| **3035** | 3036 | 24/01/2015 | Impartus | NaN | NaN | NaN | Kaizen Private Equity | Private Equity | NaN | Series A |
| **3036** | 3037 | 25/01/2015 | Thrill App | NaN | NaN | NaN | People Group (Shaadi.com) | Private Equity | 10,00,000 | Strategic Funding |
| **3037** | 3038 | 27/01/2015 | CarDekho.com | NaN | NaN | NaN | Hillhouse Capital, Tybourne Capital | Private Equity | 5,00,00,000 | Series B |
| **3038** | 3039 | 28/01/2015 | Grabhouse.com | NaN | NaN | NaN | Kalaari Capital, Sequoia Capital | Private Equity | 25,00,000 | Series A |
| **3039** | 3040 | 29/01/2015 | Printvenue | NaN | NaN | NaN | Asia Pacific Internet Group | Private Equity | 45,00,000 | NaN |
| **3040** | 3041 | 29/01/2015 | Graphene | NaN | NaN | NaN | KARSEMVEN Fund | Private Equity | 8,25,000 | Govt backed VC Fund |
| **3041** | 3042 | 30/01/2015 | Mad Street Den | NaN | NaN | NaN | Exfinity Fund, GrowX Ventures. | Private Equity | 15,00,000 | NaN |
| **3042** | 3043 | 30/01/2015 | Simplotel | NaN | NaN | NaN | MakeMyTrip | Private Equity | NaN | Strategic Funding, Minority stake |
| **3043** | 3044 | 31/01/2015 | couponmachine.in | NaN | NaN | NaN | UK based Group of Angel Investors | Seed Funding | 1,40,000 | NaN |

```
In [64]: data.shape
```

```
Out[64]: (3044, 10)
```

```
In [65]: data.describe()
```

```
Out[65]:
                Sr No
count     3044.000000
mean      1522.500000
std        878.871435
min          1.000000
25%        761.750000
50%       1522.500000
75%       2283.250000
max       3044.000000
```

In [67]:
```
data['Date dd/mm/yyyy'][data['Date dd/mm/yyyy']=='12/05.2015'] = '12/05/2015'
data['Date dd/mm/yyyy'][data['Date dd/mm/yyyy']=='13/04.2015'] = '13/04/2015'
data['Date dd/mm/yyyy'][data['Date dd/mm/yyyy']=='15/01.2015'] = '15/01/2015'
data['Date dd/mm/yyyy'][data['Date dd/mm/yyyy']=='22/01//2015'] = '22/01/2015'
```

```
<ipython-input-67-14249c5b0153>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers
us-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  data['Date dd/mm/yyyy'][data['Date dd/mm/yyyy']=='12/05.2015'] = '12/05/2015'
<ipython-input-67-14249c5b0153>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers
us-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  data['Date dd/mm/yyyy'][data['Date dd/mm/yyyy']=='13/04.2015'] = '13/04/2015'
<ipython-input-67-14249c5b0153>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers
us-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  data['Date dd/mm/yyyy'][data['Date dd/mm/yyyy']=='15/01.2015'] = '15/01/2015'
<ipython-input-67-14249c5b0153>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers
us-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  data['Date dd/mm/yyyy'][data['Date dd/mm/yyyy']=='22/01//2015'] = '22/01/2015'
```

In [72]:
```
totalStartupCount = sum(data['Startup Name'].value_counts())
print("Total Number of startUps:", totalStartupCount)
```

```
Total Number of startUps: 3044
```

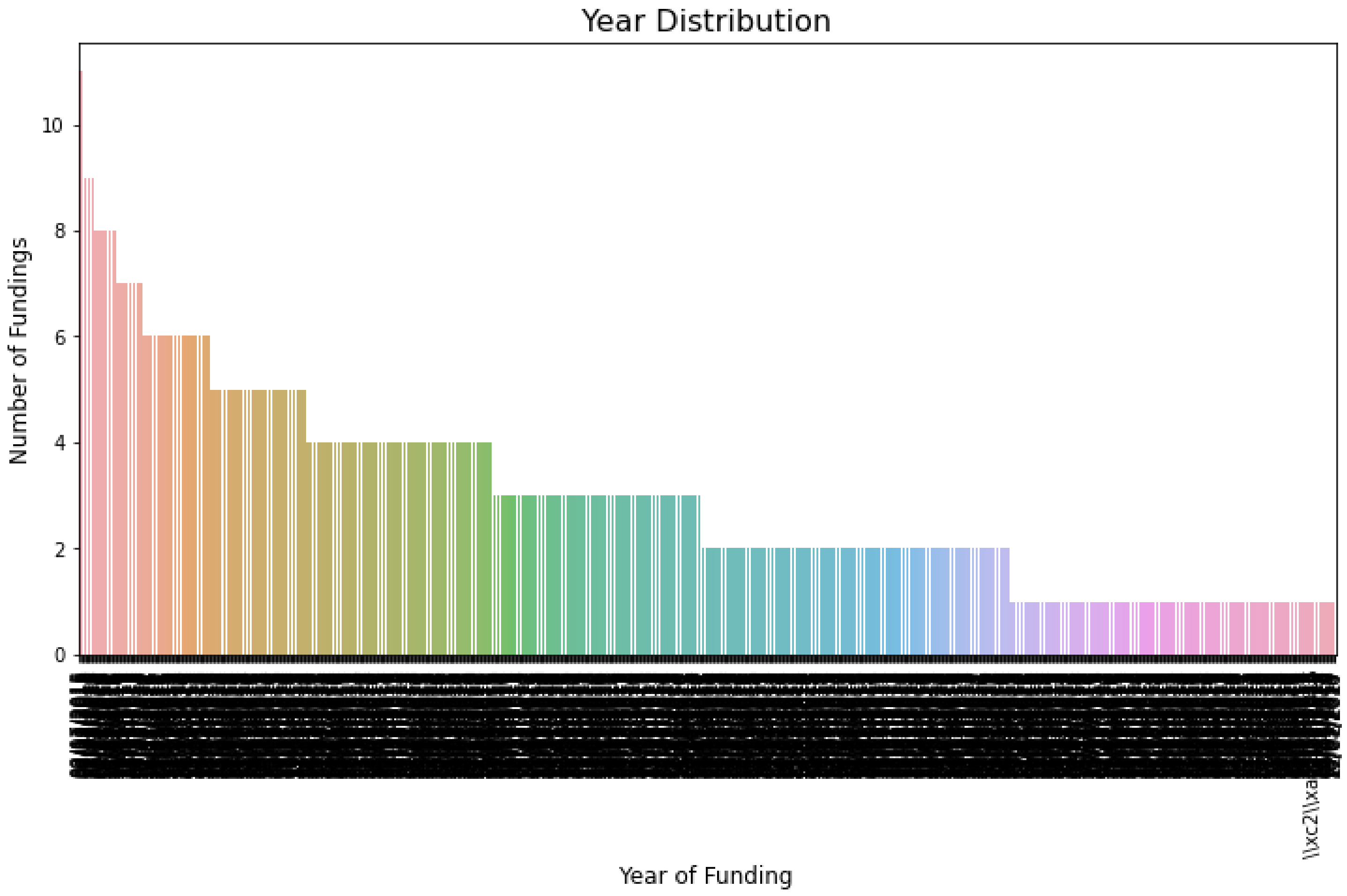
## EDA on this dataset

```
In [76]: #lets check first in which year how many funding is received by all startups together
         yearCount = data['Date dd/mm/yyyy'].value_counts()
         print(yearCount)
```

```
08/07/2015    11
02/02/2015    11
30/11/2016    11
04/10/2016    10
29/01/2016     9
              ..
09/01/2017     1
13/03/2018     1
18/03/2016     1
19/07/2018     1
22/01/2017     1
Name: Date dd/mm/yyyy, Length: 1032, dtype: int64
```

```
In [84]: plt.figure(figsize=(12,6))
         sns.barplot(yearCount.index, yearCount.values, alpha=0.8)
         plt.xticks(rotation='vertical')
         plt.xlabel('Year of Funding', fontsize=12)
         plt.ylabel('Number of Fundings', fontsize=12)
         plt.title("Year Distribution", fontsize=16)
         plt.show()
```

```
C:\Users\Radhakrishnan Nair\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as
keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

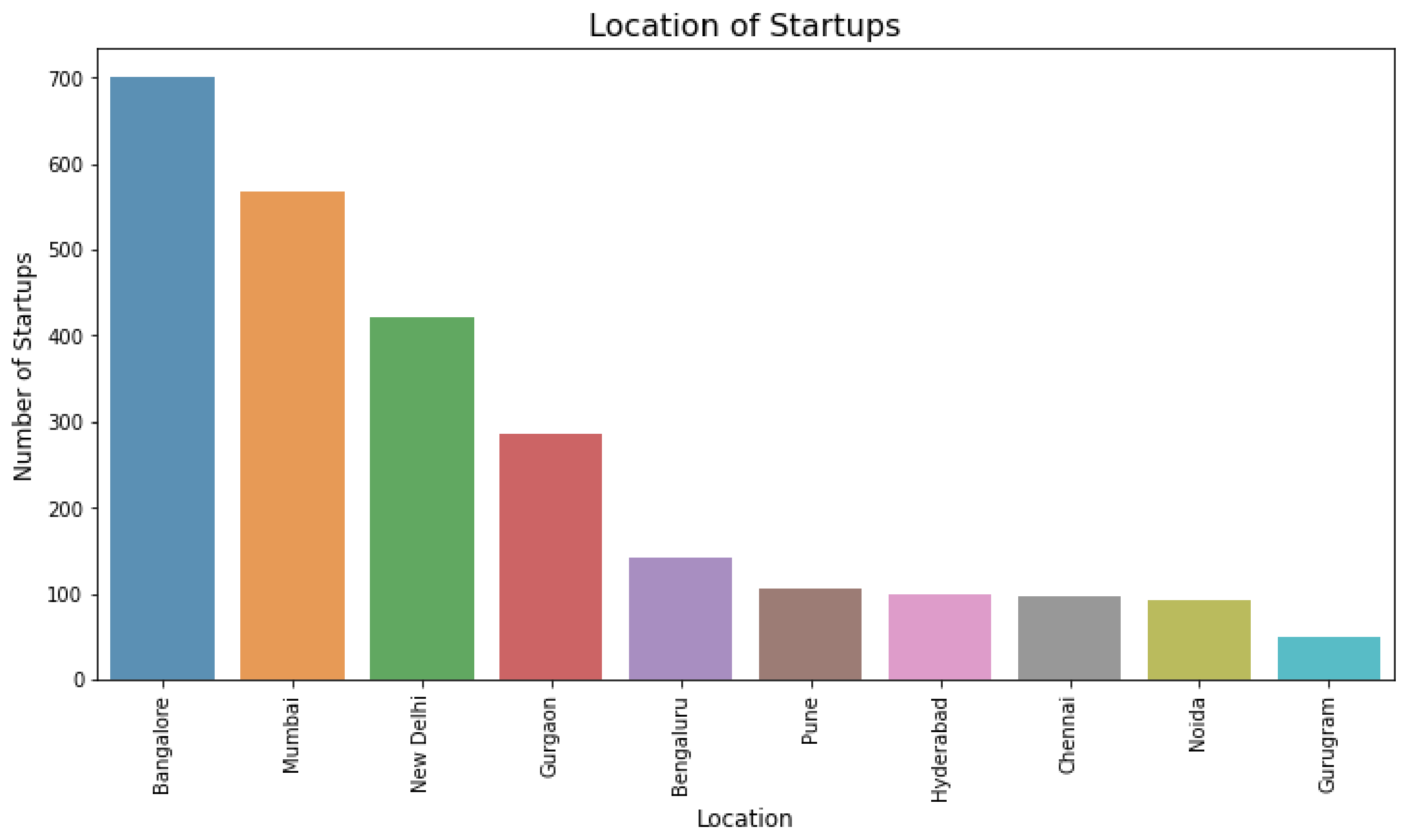


```
In [89]: #Now lets check which location recieved higest startup
         cityCount = data['City  Location'].value_counts()[:10]
         print(cityCount)
```

```
Bangalore    700
Mumbai       567
New Delhi    421
Gurgaon      287
Bengaluru    141
Pune         105
Hyderabad     99
Chennai       97
Noida         92
Gurugram      50
Name: City  Location, dtype: int64
```

```
In [90]: plt.figure(figsize=(12,6))
         sns.barplot(cityCount.index, cityCount.values, alpha=0.8)
         plt.xticks(rotation='vertical')
         plt.xlabel('Location', fontsize=12)
         plt.ylabel('Number of Startups', fontsize=12)
         plt.title("Location of Startups", fontsize=16)
         plt.show()
```

```
C:\Users\Radhakrishnan Nair\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as
keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



- We can see here that Banglore the electronic city of india is hot favourite for startups
- Difference between Banglore and Mumbai is having huge number of difference compare to difference between other cities

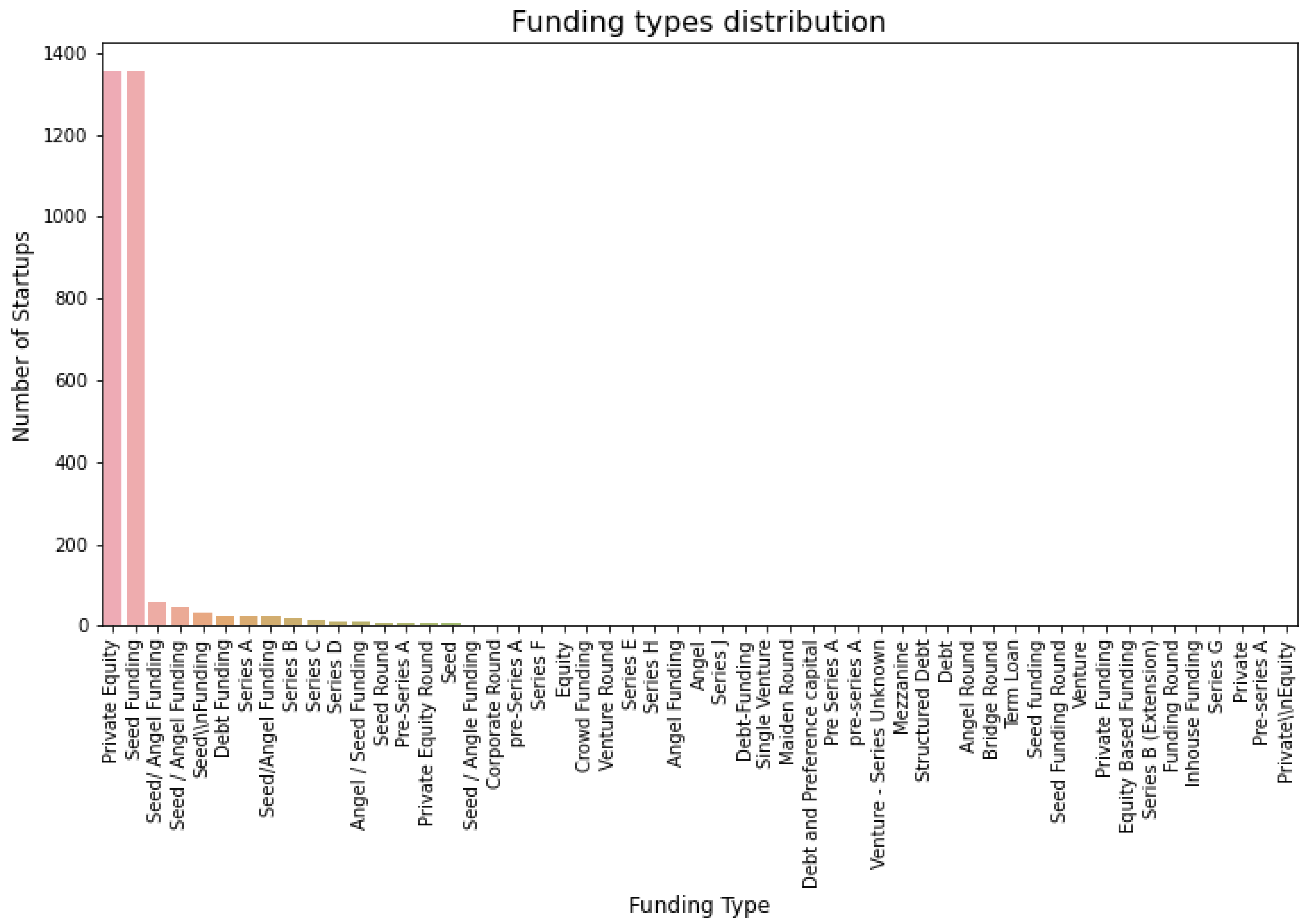
In [92]:
```
data['InvestmentnType'][data['InvestmentnType']=='SeedFunding'] = 'Seed Funding'
data['InvestmentnType'][data['InvestmentnType']=='PrivateEquity'] = 'Private Equity'
data['InvestmentnType'][data['InvestmentnType']=='Crowd funding'] = 'Crowd Funding'
investTypes = data['InvestmentnType'].value_counts()
print(investTypes)
```

```
Private Equity                1357
Seed Funding                  1355
Seed/ Angel Funding             60
Seed / Angel Funding            47
Seed\\nFunding                  30
Debt Funding                    25
Series A                        24
Seed/Angel Funding              23
Series B                        20
Series C                        14
Series D                        12
Angel / Seed Funding             8
Seed Round                       7
Pre-Series A                     4
Private Equity Round             4
Seed                             4
Seed / Angle Funding             3
Corporate Round                  2
pre-Series A                     2
Series F                         2
Equity                           2
Crowd Funding                    2
Venture Round                    2
Series E                         2
Series H                         1
Angel Funding                    1
Angel                            1
Series J                         1
Debt-Funding                     1
Single Venture                   1
Maiden Round                     1
Debt and Preference capital      1
Pre Series A                     1
pre-series A                     1
Venture - Series Unknown         1
Mezzanine                        1
Structured Debt                  1
Debt                             1
Angel Round                      1
```

```
Bridge Round                 1
Term Loan                    1
Seed funding                 1
Seed Funding Round           1
Venture                      1
Private Funding              1
Equity Based Funding         1
Series B (Extension)         1
Funding Round                1
Inhouse Funding              1
Series G                     1
Private                      1
Pre-series A                 1
Private\\nEquity             1
Name: InvestmentnType, dtype: int64

<ipython-input-92-abab9ef6bd52>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers
us-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  data['InvestmentnType'][data['InvestmentnType']=='SeedFunding'] = 'Seed Funding'
<ipython-input-92-abab9ef6bd52>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers
us-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  data['InvestmentnType'][data['InvestmentnType']=='PrivateEquity'] = 'Private Equity'
<ipython-input-92-abab9ef6bd52>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers
us-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  data['InvestmentnType'][data['InvestmentnType']=='Crowd funding'] = 'Crowd Funding'
```

In [94]:
```
plt.figure(figsize=(12,6))
sns.barplot(investTypes.index, investTypes.values, alpha=0.8)
plt.xticks(rotation='vertical')
plt.xlabel('Funding Type', fontsize=12)
plt.ylabel('Number of Startups', fontsize=12)
plt.title("Funding types distribution", fontsize=16)
plt.show()
```

```
C:\Users\Radhakrishnan Nair\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as
keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



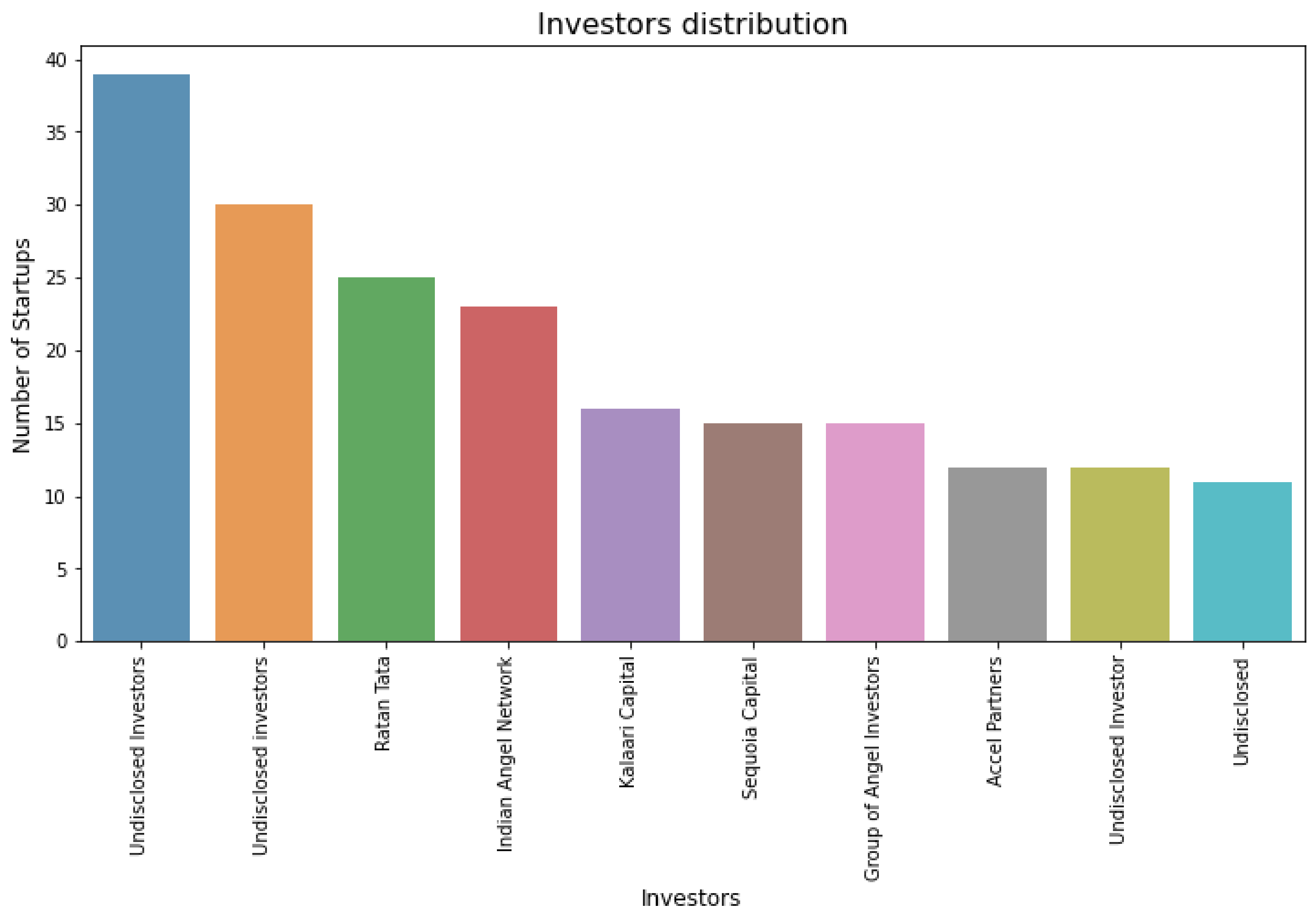
**Seed Funding and Private Equity funding are the two major funding types of investment in startups**

```
In [96]: investorCounts = data['Investors Name'].value_counts()[:10]
         print(investorCounts)
```

```
Undisclosed Investors       39
Undisclosed investors       30
Ratan Tata                  25
Indian Angel Network        23
Kalaari Capital             16
Sequoia Capital             15
Group of Angel Investors    15
Accel Partners              12
Undisclosed Investor        12
Undisclosed                 11
Name: Investors Name, dtype: int64
```

```
In [97]: plt.figure(figsize=(12,6))
         sns.barplot(investorCounts.index, investorCounts.values, alpha=0.8)
         plt.xticks(rotation='vertical')
         plt.xlabel('Investors', fontsize=12)
         plt.ylabel('Number of Startups', fontsize=12)
         plt.title("Investors distribution", fontsize=16)
         plt.show()
```

```
C:\Users\Radhakrishnan Nair\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as
keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



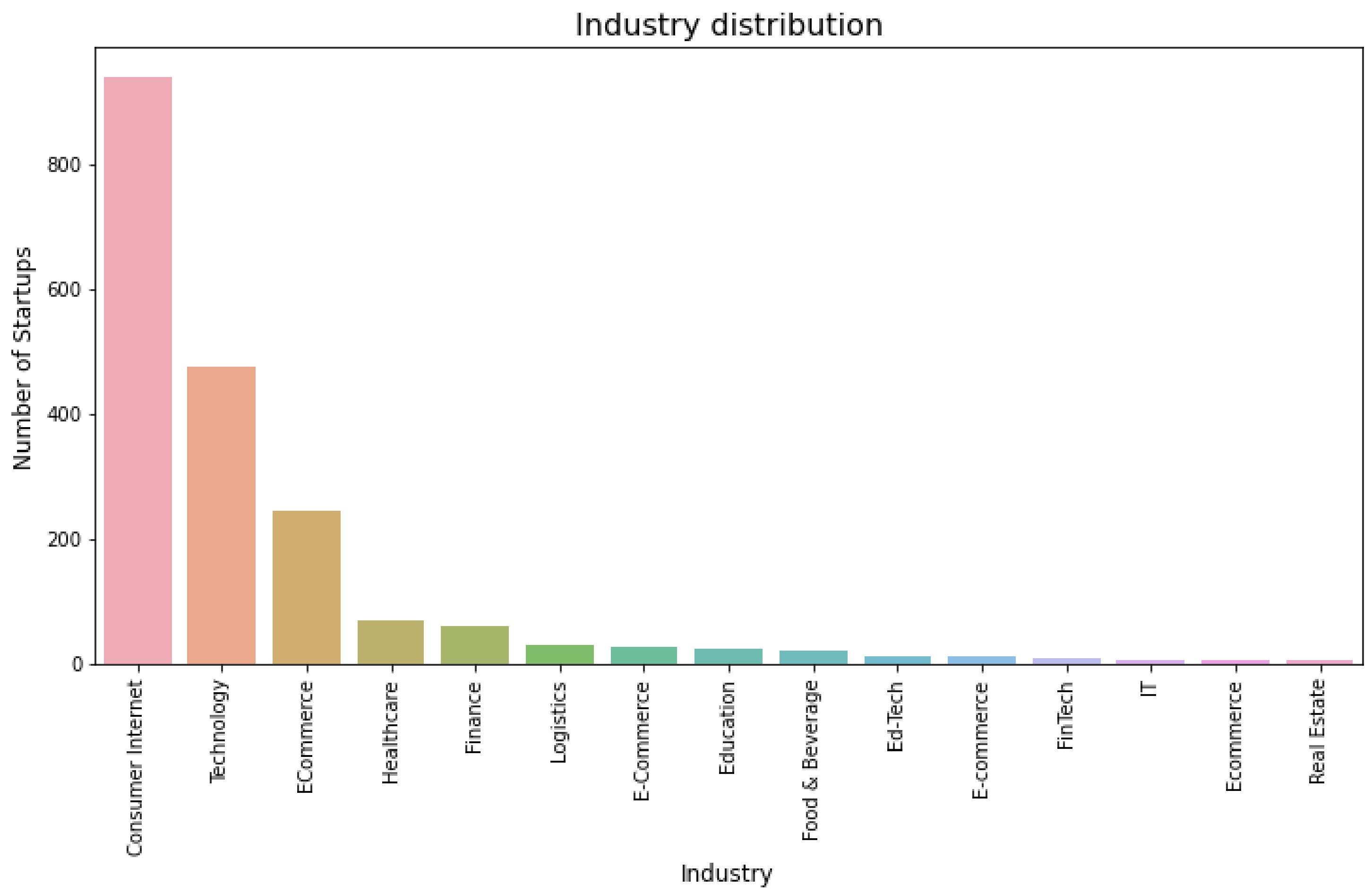
We have some ambiguity here in name 'Undisclosed Investors' we need to remove this duplication from our dataset we'll process this attribute and make all 'Undisclosed Investors' as same

```
In [99]: data['Industry Vertical'][data['Industry Vertical']=='eCommerce'] = 'ECommerce'
         industryCounts = data['Industry Vertical'].value_counts()[:15]
         print(industryCounts)
```

```
Consumer Internet    941
Technology           478
ECommerce            247
Healthcare            70
Finance               62
Logistics             32
E-Commerce            29
Education             24
Food & Beverage       23
Ed-Tech               14
E-commerce            12
FinTech                9
IT                     8
Ecommerce              8
Real Estate            6
Name: Industry Vertical, dtype: int64

<ipython-input-99-24a1da23a2b3>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers
us-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  data['Industry Vertical'][data['Industry Vertical']=='eCommerce'] = 'ECommerce'
```

```
In [100]: plt.figure(figsize=(12,6))
          sns.barplot(industryCounts.index, industryCounts.values, alpha=0.8)
          plt.xticks(rotation='vertical')
          plt.xlabel('Industry', fontsize=12)
          plt.ylabel('Number of Startups', fontsize=12)
          plt.title("Industry distribution", fontsize=16)
          plt.show()
```

```
C:\Users\Radhakrishnan Nair\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as
keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



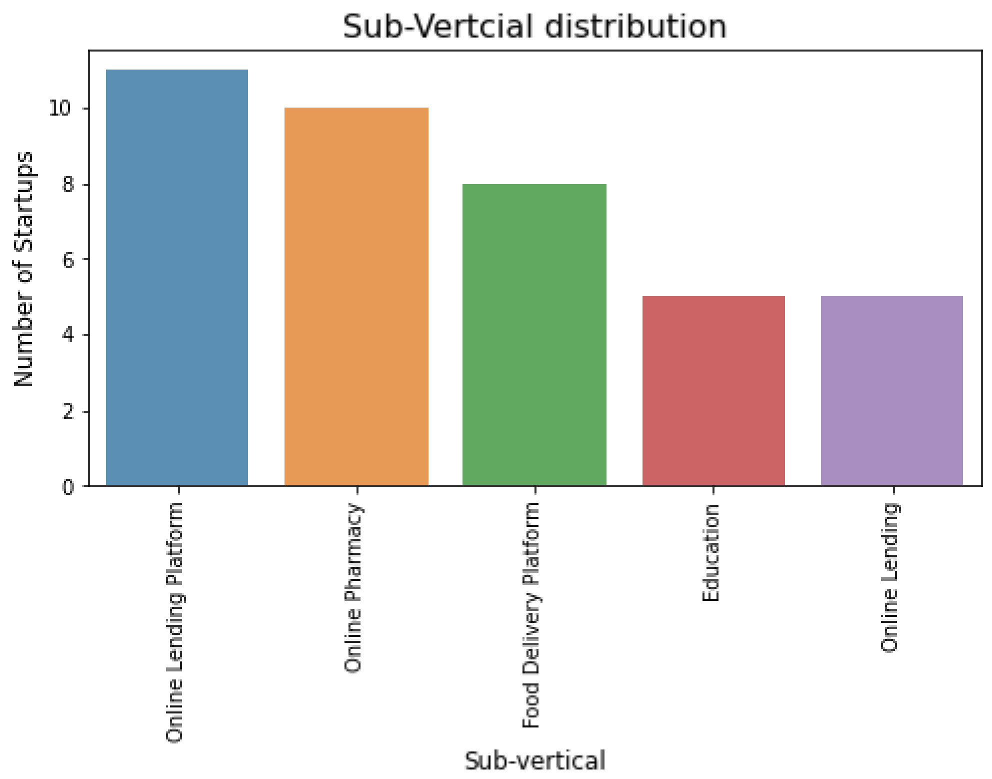
- Majority of startups are working in 'Consumer Internet'
- 2nd heighest that is 'technology' some startups interested in developing technologies

```
In [105]: subVerticalCounts = data['SubVertical'].value_counts()[:5]
          print(subVerticalCounts)
```

```
Online Lending Platform    11
Online Pharmacy            10
Food Delivery Platform      8
Education                   5
Online Lending              5
Name: SubVertical, dtype: int64
```

```
In [107]: plt.figure(figsize=(8,4))
          sns.barplot(subVerticalCounts.index, subVerticalCounts.values, alpha=0.8)
          plt.xticks(rotation='vertical')
          plt.xlabel('Sub-vertical', fontsize=12)
          plt.ylabel('Number of Startups', fontsize=12)
          plt.title("Sub-Vertcial distribution", fontsize=16)
          plt.show()
```

```
C:\Users\Radhakrishnan Nair\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as
keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



- Online Pharmacy is the main concentartion followed by Food Delivery platform for new startups

```
In [108]: data.columns
```

```
Out[108]: Index(['Sr No', 'Date dd/mm/yyyy', 'Startup Name', 'Industry Vertical',
                 'SubVertical', 'City  Location', 'Investors Name', 'InvestmentnType',
                 'Amount in USD', 'Remarks'],
                dtype='object')
```

## Outputs

- estimating the percentage error to find the best baseline model
- Percentage of people who got promoted from each department
- Distribution of promotions among people with different Educational backgrounds
- Mean score of employees with different educational background
- Filling the missing values

## Final EDA

Q.Which are the best Cities for Startups in India?

A. Banglore the electronic city of india is hot favourite for startups followed by Mumbai and New Delhi.

Q.Which type of funding is done in India for startups Ecosystem?

A.Seed Funding and Private Equity funding are the two major funding types of investment in startups

Q.Which are the Top domains where new startups are Imerging?

A.Majority of startups are working in Consumer Internet,

2nd heighest that is technology some startups interested in developing new technologies

Q. Which types of services are the main concenration for new satrtups?

A.Online Pharmacy is the main concentartion followed by Food Delivery platform for new startups