

**THAPAR INSTITUTE OF ENGINEERING
AND
TECHNOLOGY, PATIALA**

UCS538 (Fundamentals of Data Science)

Dashboard Using Python

(<https://startupsurvey.herokuapp.com/>)



NAME: Bhavesh Sareen

ROLL NO: 101917085

BATCH: 3CSE4

EMAIL ID: bsareen_be19@thapar.edu

PHONE NO: +91 7837303852

Indian Startups Funding Dashboard

Code (with Comments)

```
# importing all the libraries needed for the project
import pandas as pd
from pandas._config.config import options
import plotly.express as px
import streamlit as st
import numpy as np

# setting the page title
st.set_page_config(page_title="Startup Funding Dashboard ",
                    page_icon=":moneybag:",
                    layout="wide",
                    )

# reading the csv file
dataf = pd.read_csv("startup_funding.csv",
                    skipinitialspace=True,
                    )

# Preprocessing Steps

# Correcting the name of the City in 'CityLocation' column
dataf['CityLocation'].replace('Delhi', 'New Delhi', inplace=True)
dataf['CityLocation'].replace('bangalore', 'Bangalore', inplace=True)

# Getting all the first City name
dataf['CityLocation'] = dataf['CityLocation'].str.split(' /').str[0]

# Correcting the investment type
dataf['InvestmentType'].replace('SeedFunding', 'Seed Funding', inplace=True)
dataf['InvestmentType'].replace('Crowd funding', 'Crowd Funding',
inplace=True)
dataf['InvestmentType'].replace(
    'PrivateEquity', 'Private Equity', inplace=True)

# dropping the null values
dataf.dropna(subset=['InvestmentType'], inplace=True)

# Correcting the industry(domain) of the startup
dataf1 = dataf.copy()
```

```

dataf1.dropna(subset=['IndustryVertical', 'AmountInUSD'], inplace=True)
dataf1['IndustryVertical'].replace('ecommerce', 'Ecommerce', inplace=True)
dataf1['IndustryVertical'].replace('ECommerce', 'Ecommerce', inplace=True)
dataf1['IndustryVertical'].replace('eCommerce', 'Ecommerce', inplace=True)

# function for Converting the amount string into integer and removing the
starting and ending spaces by using strip() function
def amount_generator(amt):
    amt = amt.split(',')
    ans = ""
    for i in range(len(amt)):
        ans = ans + (amt[i].strip())
    return int(ans)

dataf1['AmountInUSD'] = dataf1['AmountInUSD'].apply(amount_generator)

# Mainpage Starts
st.title(":chart_with_upwards_trend: Indian Startups Funding Dashboard")
st.markdown("###")

# Building the subheading section
total_amt = (dataf1['AmountInUSD'].sum())
total_amt = (total_amt.astype(float)/1000000000).round(2).astype(str) + 'B'
lcol, midcol, rightcol = st.columns(3)
with lcol:
    st.subheader("Total Amount Funded:")
    st.subheader(f"US$ {total_amt}")
# st.markdown for leave line
st.markdown("###")

# If user wants to see the dataset
agree = st.checkbox('See Dataset')
if agree:
    st.dataframe(dataf)

# Displaying the line chart
line_chart_data = dataf.copy()
# As year column contains data in dd/mm/yyyy to fetch only the yyyy I applied
the slicing on it
line_chart_data['Year'] = line_chart_data['Date'].str[-4:]

# Calculating the year count ex that how many times 2016 comes in the column
for this applied value_counts function on Year column
year = line_chart_data.Year.value_counts()
year.sort_index(inplace=True)

# Used plotly inbuilt function px.line which is used to plot the line graph

```

```

line_chart = px.line(year)
# Adding the styles to the line graph
line_chart.update_layout(title='<b>Startups Funding Vs Year<b>',
                          xaxis_title='<b>Years<b>',
                          yaxis_title='<b>No. of Fundings<b>',
                          font_size=16,
                          )

# For the Alignment purpose divide the width of the browser screen into left
and right columns
left_column, right_column = st.columns(2)
left_column.plotly_chart(line_chart, use_container_width=True)

# plotting the pie graph which is displaying the different values of a various
Industries(domain) of startups (e.g. percentage distribution)

# converting the IndustryVertical and AmountInUSD column to lists
indlist = list(dataf1.IndustryVertical)
amt = list(dataf1.AmountInUSD)

# here d(dictionary) storing the industry as a key and values stores the total
amount given to that industry
d = {}

# claculating the amt corresponding to that industry
for i in range(len(indlist)):
    # I have passed 0 in the d.get so that if key does not exist in the dict
it returns 0 value
    d[indlist[i]] = d.get(indlist[i], 0)+amt[i]

# Converting the dict keys and values into the list
key = list(d.keys())
values = list(d.values())

# Making the np array
key = np.array(key)
values = np.array(values)

# apply the argsort function to the values np array which returns indices of
the sorted values
ind = values.argsort()[::-1]
# taking only the first 10 indices
ind = ind[:10]

# getting the industry and amount corresponding to the indices
key = key[ind]
values = values[ind]

```

```

# For percentage distribution calculating the percentages amt of industries in
which startups belongs
values = (values/sum(values))*100
# Rounding to 2 decimal values
values = np.round(values, decimals=2)

# plotting the pie graph
pie_chart = px.pie(
    values=values,
    names=key,
    title="<b>Top 10 Industries in terms of funding </b>",
)
# colors = ['#890DA2', '#FA9D3B', '#C03B83', '#DE6164', '#EF7E50']
colors = ['rgb(33, 75, 99)', 'rgb(79, 129, 102)',
          'rgb(151, 179, 100)', 'rgb(175, 49, 35)', 'rgb(36, 73, 147)']
# Styling the Pie graph
pie_chart.update_traces(hoverinfo='label+percent', textposition='inside',
                        textinfo='percent',
                        textfont_size=20, marker=dict(colors=colors,
line=dict(color='#000000', width=2)))
pie_chart.update_layout(uniformtext_minsize=12, uniformtext_mode='hide',
font_size=16, annotations=[dict(
    text='', x=0.18, y=0.5, font_size=20, showarrow=False), dict(text='',
x=0.82, y=0.5, font_size=20, showarrow=False)])

right_column.plotly_chart(pie_chart, use_container_width=True)

# plotting th bar graph between location and number of fundings to that
location

st.markdown("###")
# Giving option to the user to choose the investment_type
investment_type = st.multiselect(
    "Select the Investment Type: ",
    options=dataf["InvestmentType"].unique(),
    default=dataf["InvestmentType"].unique()
)
df_selection = dataf.query(
    "InvestmentType == @investment_type"
)

city_number = df_selection['CityLocation'].value_counts()[0:10]
city = city_number.index
numCity = city_number.values

# Plotting the bar graph
bar_graph = px.bar(city_number,
                    text=city_number.values,

```

```

        orientation="v",
        width=1100,
        template="plotly_dark")

# Styling the Graph
bar_graph.update_traces(marker_color=['#0D0887', '#41039D', '#6F01A7',
'#890DA2', '#A01B9B', '#C03B83',
                                '#DE6164', '#EF7E50', '#FA9D3B', '#FCB331'],
marker_line_color='cyan', marker_line_width=1.5, opacity=1)
bar_graph.update_traces(texttemplate='%{text:.2s}', textposition='outside')
bar_graph.update_yaxes(range=[0, 700])
st.markdown("###")
bar_graph.update_layout(
    title_text='<b>Top 10 Indian Cities location for startup<b>',
    xaxis_title='<b>Location<b>',
    yaxis_title='<b>No. of Fundings<b>',
    margin=dict(t=30, l=380, r=50, b=50),
    font_size=16,)
# Showing the bar graph
st.write(bar_graph, use_container_width=True)

# Creating the Pie graph for the
# The list of top investors from dataset
# Creating the Dictionary in which we store the key as name of the investors
and
# value as the number of times they invested

dataf2 = dataf.copy()
# Preprocessing
dataf2.dropna(subset=['InvestorsName'], inplace=True)

# Function for creating the dict

def make_dictionary(arr):
    dictionary = {}
    for i in arr:
        # Dataset have undisclosed investors and I do not take the Undisclosed
        Investors and also the investors with blank name in the consideration
        if i == 'Undisclosed Investors' or i == 'Undisclosed investors' or i
        == '':
            continue
        # In InvestorsName column it contains multiple investors separated
        with , SO I have taken care of this also
        if ',' not in i:
            if i in dictionary:
                dictionary[i] = dictionary.get(i)+1
            else:

```

```

        dictionary[i] = 1
    else:
        string = i.strip().split(',')
        for j in string:
            if j.strip() in dictionary:
                dictionary[j.strip()] = dictionary.get(j.strip())+1
            elif j.strip() != "":
                dictionary[j.strip()] = 1
    return dictionary

st.markdown("###")
# Giving Option to the user to select the number of investors
investor_selector = st.slider('Select the numbers of Investors you want:',
                               min_value=2,
                               max_value=15,
                               value=(2, 15)
                               )
# Giving Option to the user to select the investment type of the investor
invest_type = st.multiselect(
    "Select the Investment Type of the Investor: ",
    options=dataf2["InvestmentType"].unique(),
    default=dataf2["InvestmentType"].unique()
)
st.markdown("###")
df_select = dataf2.query(
    "InvestmentType == @invest_type"
)

# Getting the Dictionary
dictionary = make_dictionary(df_select['InvestorsName'])

# Converting keys and values to the list
key1 = list(dictionary.keys())
values1 = list(dictionary.values())
inv1 = len(key1)

# Displaying the subheading at the top of the mainpage
with midcol:
    st.subheader("Total No. of Investors:")
    st.subheader(f"{inv1}")

# Converting to the numpy array
key1 = np.array(key1)
values1 = np.array(values1)
ind = values1.argsort()[::-1]

# investor_selector stores the tuple which is the range selected by the user
at the frontend and I am fetching the first element of the tuple

```

```

ind = ind[:investor_selector[0]]

key1 = key1[ind]
values1 = values1[ind]
values1 = (values1/sum(values1))*100
values1 = np.round(values1, decimals=2)

# converting into the dictionary
d1 = {
    'iname': key1,
    'ivalue': values1
}
# Converting into the Datframe which is used to plot the pie_chart
df1 = pd.DataFrame(d1)
pie_chart1 = px.pie(
    data_frame=df1,
    values='ivalue',
    template='presentation',
    names='iname',
    title=f'<b>Top {investor_selector[0]} Investors names who have invested
maximum number of times<b>',
    width=1100,
    height=500,
    hole=0.4,
)

colors = ['rgb(33, 75, 99)', 'rgb(79, 129, 102)',
          'rgb(151, 179, 100)', 'rgb(175, 49, 35)', 'rgb(36, 73, 147)']

# Applied Styling
pie_chart1.update_traces(hoverinfo='label+percent', textposition='inside',
textinfo='percent', textfont_size=20,
                        marker=dict(colors=colors, line=dict(color='#000000',
width=2)))
pie_chart1.update_layout(uniformtext_minsize=12,
                        uniformtext_mode='hide',
                        font_size=16,
                        margin=dict(t=30, l=380, r=50, b=50),
                        annotations=[dict(text='', x=0.18, y=0.5,
font_size=20, showarrow=False), dict(text='', x=0.82, y=0.5, font_size=20,
showarrow=False)])
st.markdown("##")
# Displaying the graph
st.write(pie_chart1)

# Calculations for plotting the bar graph of the top startups based on fundings
received

```



```

dataf3 = dataf.copy()
# Preprocessing Steps
# Here I have replaced the different names of the same startup to the one name
dataf3 .dropna(subset=['StartupName', 'AmountInUSD'], inplace=True)
dataf3['StartupName'].replace('Flipkart.com', 'Flipkart', inplace=True)
dataf3['StartupName'].replace('Ola Cabs', 'Ola', inplace=True)
dataf3['StartupName'].replace('OlaCabs', 'Ola', inplace=True)
dataf3['StartupName'].replace('Ola Cabs', 'Ola', inplace=True)
dataf3['StartupName'].replace('OlaCabs', 'Ola', inplace=True)
dataf3['StartupName'].replace('Oyo Rooms', 'Oyo', inplace=True)
dataf3['StartupName'].replace('OyoRooms', 'Oyo', inplace=True)
dataf3['StartupName'].replace('OYO Rooms', 'Oyo', inplace=True)
dataf3['StartupName'].replace('Paytm Marketplace', 'Paytm', inplace=True)
dataf3['AmountInUSD'] = dataf3['AmountInUSD'].apply(amount_generator)

# Converting into the list
st1 = list(dataf3.StartupName)
amt = list(dataf3.AmountInUSD)
d = {}
slen = len(st1)
# For subheading section at the top of the main page
with rightcol:
    st.subheader("Total Startups incorporated:")
    st.subheader(f"{slen}")

for i in range(len(st1)):
    d[st1[i]] = d.get(st1[i], 0)+amt[i]

# Applied the same steps as applied earlier for sorting
key2 = list(d.keys())
values2 = list(d.values())
key2 = np.array(key2)
values2 = np.array(values2)
ind = values2.argsort()[::-1]
ind = ind[:40]
key2 = key2[ind]
values2 = values2[ind]
for i in range(len(values2)):
    d[key2[i]] = values2[i]

# Creating the dictionary having the key as startup name and value as total
amt funded to it
the_dict = {'sname': key2, 's_amt': values2}

bar_graph2 = px.bar(the_dict, x='sname', y='s_amt',
                    orientation="v",
                    width=1100,
                    height=500,

```

```

        template="plotly_dark")

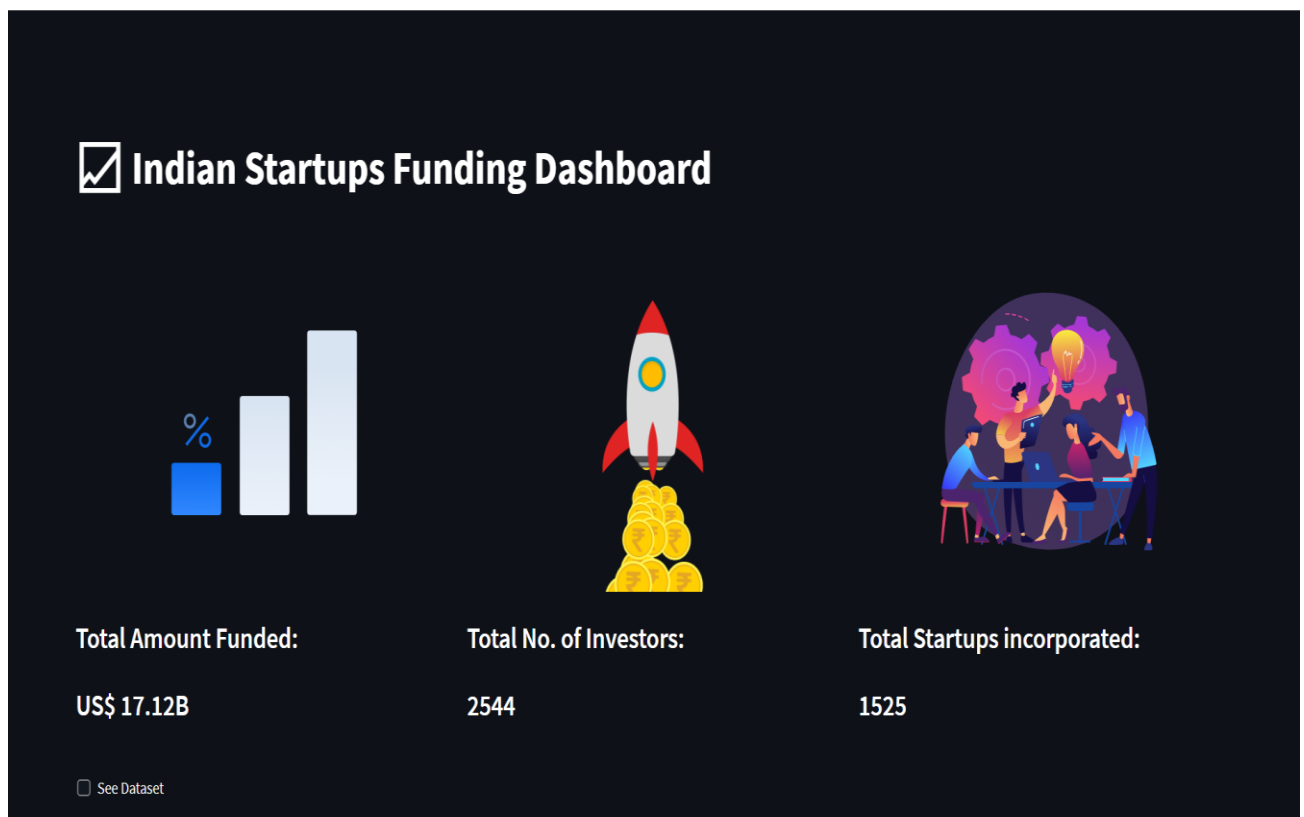
# For Styling
colors1 = ['#0D0887', '#41039D', '#6F01A7', '#890DA2', '#A01B9B',
           '#C03B83', '#DE6164', '#EF7E50', '#FA9D3B', '#FCB331',
           '#0D0887', '#41039D', '#6F01A7', '#890DA2', '#A01B9B',
           '#C03B82', '#DE6165', '#EF7E52', '#FA9D6B', '#FCB339',
           '#0D0887', '#41039D', '#6F01A7', '#890DA2', '#A01B9B',
           '#C03B83', '#DE6164', '#EF7E50', '#FA9D3B', '#FCB331',
           '#0D0887', '#41039D', '#6F01A7', '#890DA2', '#A01B9B',
           '#C03B83', '#DE6164', '#EF7E50', '#FA9D3B', '#FCB331',
           '#0D0887', '#41039D', '#6F01A7', '#890DA2', '#A01B9B',
           '#C03B83', '#DE6164', '#EF7E50', '#FA9D3B', '#FCB331',
           '#0D0887', '#41039D', '#6F01A7', '#890DA2', '#A01B9B',
           '#C03B83', '#DE6164', '#EF7E50', '#FA9D3B', '#FCB331']*5
bar_graph2.update_traces(marker_color=colors1, marker_line_color='cyan',
                          marker_line_width=0.5, opacity=1)

st.markdown("##")
bar_graph2.update_layout(
    title_text='<b>Top Startups based on Fundings<b>',
    xaxis_title='<b>Startup Name<b>',
    yaxis_title='<b>Amount<b>',
    margin=dict(t=30, l=380, r=50, b=50),
    font_size=16,
)
st.write(bar_graph2)

# This line of statements is used to hide the prebuilt styling in the
streamlit
hide_st_style = """
    <style>
    #MainMenu {visibility: hidden;}
    footer {visibility: hidden;}
    header {visibility: hidden;}
    </style>
    """
st.markdown(hide_st_style, unsafe_allow_html=True)

```

Screenshots:



Landing Page of the Dashboard

Added Some Animations using Lottie animations library.

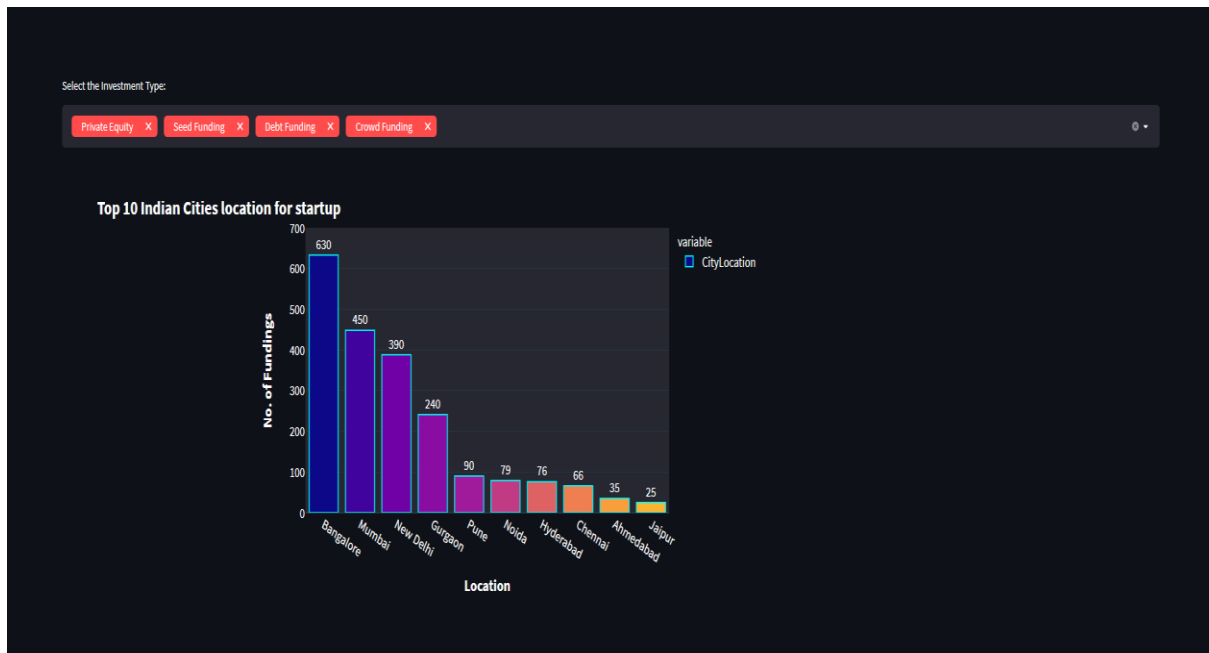
Also showing some statistics related to the Startups Fundings

- Total Amount Funded
- Total No. of Investors
- Total Startups incorporated



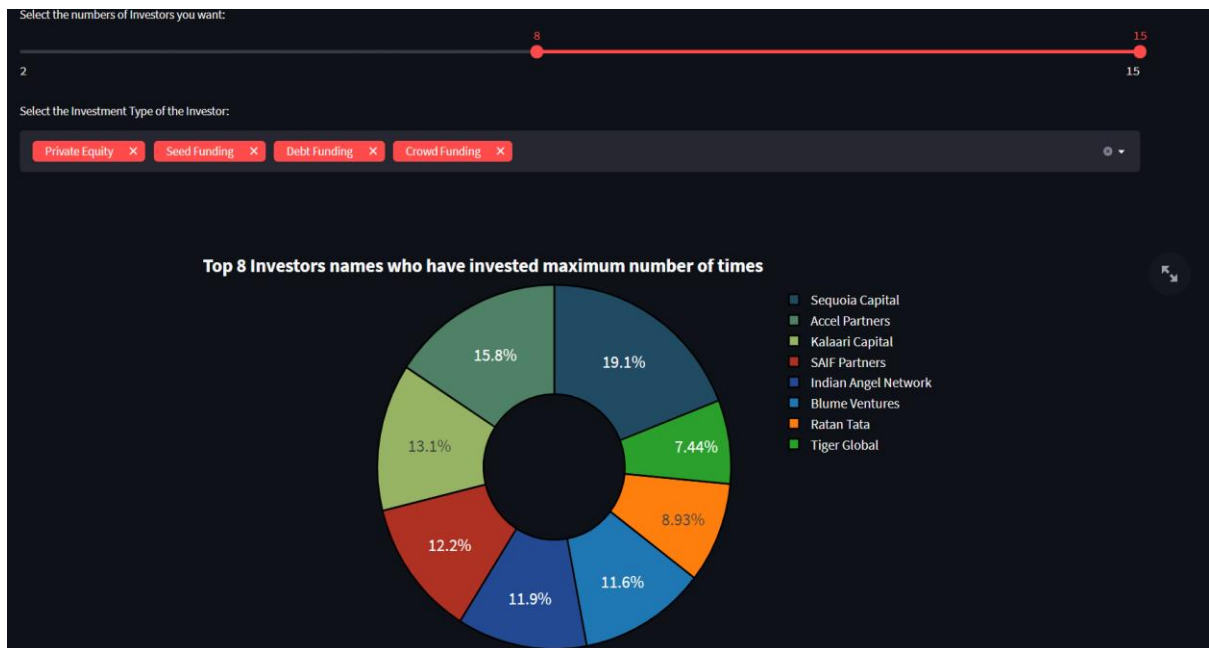
Summary:

- This screenshot contains line graph on the left side which is plotted between the Year vs No of Fundings in that year.
- In line graph we can see that there is decline in the total number of fundings in 2017.
- On the right there is pie graph which is representing the percentage distribution of the Top 10 Industries in terms of fundings
- We can depict that Ecommerce industry has highest number of fundings



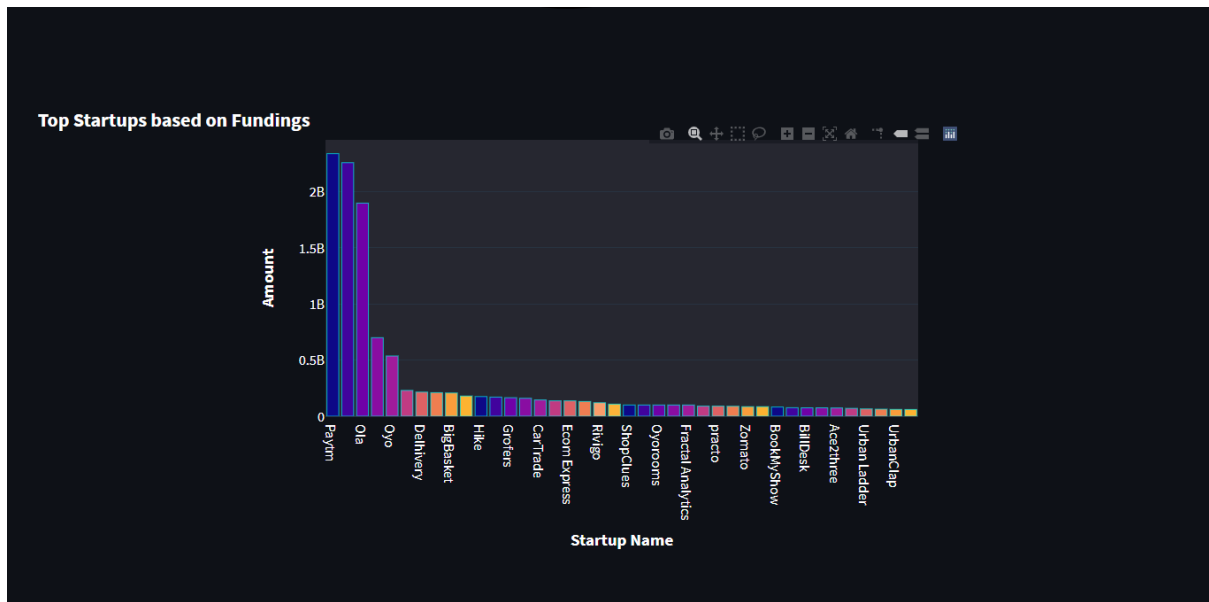
Summary:

- Here I have plotted the bar graph which represents the total number of fundings startups got in the top 10 Indian Cities Locations
- Visualising this graph helps us to search for a perfect location where getting the investment has a high chance.
- Ex: Bangalore has high chances of getting the investment



Summary:

- Here I have plotted the pie graph which represents the percentage distribution of the Top investors who have invested maximum number of times.
- User can slide the slider and select the number of investors for which the user wants to see the percentage distribution.
- This percentage distribution helps the startups for getting the initial investment by contacting these investors.



Summary:

- Here I plotted the bar graph which gives us the total amount received by the startup.
- Paytm received the highest amount of fundings.

URL

I have also hosted this dashboard on Internet using the Heroku hosting services

Link:

<https://startupsurvey.herokuapp.com/>

References

- <https://docs.streamlit.io/>
- <https://plotly.com/python/>
- <https://www.kaggle.com/sudalairajkumar/indian-startup-funding>
- <https://lottiefiles.com/search?q=Buisness&category=animations>
- <https://www.webfx.com/>
- <https://id.heroku.com/login>

Dataset taken from

- <https://www.kaggle.com/sudalairajkumar/indian-startup-funding>

Dashboard URL

<https://startupsurvey.herokuapp.com/>