STARTUP PROPHET

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(DS)

Submitted By

SALLA SAHAS	**21UK1A6704**
GATTIKOPPULA MANIKUMAR	**21UK1A6758**
SUDHATI KIRAN TEJA	**21UK1A6753**
AKARAPU JYOTHI	**21UK1A6743**

Under the guidance of

Dr.K. SHARMILA REDDY

(Hod of CSE(Data Science))

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(DS)

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

VAAGDEVI ENGINEERING COLLEGE(WARANGAL)

CERTIFICATE OF COMPLETION
INDUSTRY ORIENTED MINI PROJECT

This is to certify that the UG Project Phase-1 entitled “STARTUP PROPHET” is being submitted by SAHAS SALLA(21UK1A6704), GK MANI(21UK1A6758), AKARAPU JYOTHI(21UK1A6743),SUDHATI KIRAN(21UK1A6753) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2023- 2024.

Project Guide	HOD
Dr.K. SHARMILA REDDY	**Dr.K. SHARMILA REDDY**
(HOD of CSE(Data Science))	(HOD of CSE(Data Science))

EXTERNAL

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to our guide **Dr.K.SHARMILA**, Head of the Department of CSE(Data Science), Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to the guide, **Dr.K. SHARMILA ,** Head of department of CSE(Data Science) for her constant guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

SAHAS SALLA (21UK1A6704)
AKARAPU JYOTHI (21UK1A6743)
SUDHATI KIRAN(21UK1A6753)
GK MANI KUMAR(21UK1A6758)

ABSTRACT

The dynamic landscape of startup ventures poses significant challenges for entrepreneurs, investors, and stakeholders in predicting the success of new enterprises. "Startup Prophet" is a predictive analytics application developed to address this challenge by leveraging machine learning techniques to forecast the success probability of startups. This web-based application utilizes a machine learning model trained on diverse datasets containing historical data on startup performance, market conditions, and various other influential factors.

Startup Prophet serves as a valuable tool for entrepreneurs seeking to assess their venture's potential, investors looking to make informed funding decisions, and analysts interested in market trends. By integrating sophisticated machine learning techniques with an accessible platform, Startup Prophet aims to demystify the startup success prediction process and foster informed decision-making in the entrepreneurial ecosystem.

INDEX

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW:

A startup or start-up is a company or project begun by an entrepreneur to seek, develop, and validate a scalable economic model. While entrepreneurship refers to all new businesses, including self-employment and businesses that never intend to become registered, startups refer to new businesses that intend to grow large beyond the solo founder. Startups face high uncertainty and have high rates of failure, but a minority of them do go on to be successful and influential. Startups play a major role in economic growth. They bring new ideas, spur innovation, create employment thereby moving the economy. There has been an exponential growth in startups over the past few years. Predicting the success of a startup allows investors to find companies that have the potential for rapid growth, thereby allowing them to be one step ahead of the competition.

The objective is to predict whether a startup which is currently operating turns into a success or a failure. The success of a company is defined as the event that gives the company's founders a large sum of money through the process of M&A (Merger and Acquisition) or an IPO (Initial Public Offering). A company would be considered as failed if it had to be shut down.

1.2 PURPOSE:

The purpose of this project is to predict the startup prophet on machine learning model classification using the support vector machine algorithm and random forest model by this we can be capable to predict the startup prophet classification.

CHAPTER 2:
LITERATURE SURVEY

2.1 EXISTING PROBLEM

India is seeing a growth phase under the leadership of able people. However, there still exist many challenges that need to be addressed. To solve these challenges and problems, the country as a whole must be engaged, and talent must be brought from outside the government domain, especially where domain knowledge or entrepreneurial leadership is required. People who are passionate create great things, and companies that aspire to solve bigger problems do much better than those who just look around for funding and money. A combination of talent and diverse experiences backed by strong political will are the key ingredients to coming up with out-of-the-box solutions to address the many challenges we face as a developing country. We look at some of the real issues in India that startups can aim to address.

Instant access to healthcare One of the most critical needs today is access to good healthcare. Billions around the world, particularly people in the Indian subcontinent, struggle because they do not get proper access to healthcare. Even those with access have a sour experience. That exists apps that let us book movie tickets and seats in a jiffy or even find that perfect restaurant! However, finding doctors is still unbelievably tough. Patient records are either maintained in fat files or if they are online, they are often not accessible or understandable. Doctors do not usually have the time to go through all the reports and this may lead to a compromise on the health front. Health-based startups can address a lot of issues plaguing instant access to healthcare in India. Healthcare is undergoing a major change and smartphones will soon replace doctors for more than 80 percent of health-related problems! Public transportation In India, the pains of a city's chaotic public transport system.

2.2 PROPOSED SOLUTION:

we will prepare the data using JUPYTER notebook and we use various models to predict the output. machine learning models are used very useful in predicting outcomes for large amount datasets. We use support vector machine and random forest model machine learning algorithm to predict the startup prophet classification.

CHAPTER 3
THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM

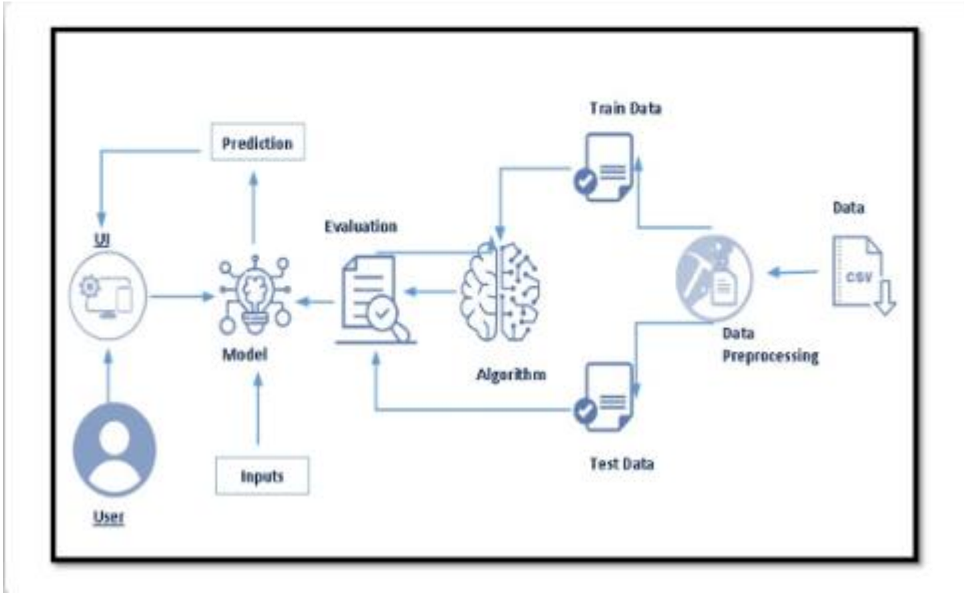


Fig 01: Flow chart

3.2 SOFTWARE & HARDWARE DESIGNING:

Software Requirements

REQUIREMENTS	SPECIFICATIONS
Anaconda Navigator	You must have anaconda installed in your device prior to begin.
➢ Anaconda prompt, google colab, Flask, spyder ➢ Frame work .	➢ One should have anaconda prompt and google colab. ➢ One should install flask framework through Anaconda prompt for running their web application. ➢ We need to build the mode; using JUPYTER notebook with all the imported packages.
Web browser	For all Web browsers, the following must be enabled: ● Cookies ● Java script

Hardware Requirements:

REQUIREMENTS	**SPECIFICATIONS**
Operating system	➢ Microsoft windows ➢ Unix ➢ Linux
Processing	Minimum: 4 CPU cores for one user. For each deployment, a sizing exercise is highly recommended.
RAM	Minimum 8 GB.
Operating system specifications	File descriptor limit set to 8192 on UNIX and Linux
Disk space	A minimum of 7 GB of free space is required to install the software.

CHAPTER 4
EXPERIMENTAL ANALYSIS

Analysis or the investigation made while working on the solution:

While working on the solution we investigated on what is Startup prophet, visualizing and analyzing the data, data pre-processing, Machine Learning service, model building. The key role on investigation is collection of data set.

DATA PRE-PROCESSING:

As we have understood how the data is let's pre-process the collected data.

The download data set is not suitable for training the machine learning model as it might have so much of randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

- → Handling missing values
- → Handling categorical data
- → Scaling Techniques
- → Handling class imbalance
- → Splitting dataset into training and test set

Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

DATA SET COLLECTION:

- ➔ Kaggle.com
- ➔ Machine learning repository

The data set contains thirteen classes:

1.is_ecommerce

2.is_otherstate

3.has_angel

4.has_roundA

5.has_roundB

6.has_roundC

7.has_roundD

8.is_top500

9.labels

10.has_VC

11.funding_rounds

12.relationships

13.milestones

CHAPTER 5:
FLOW CHART

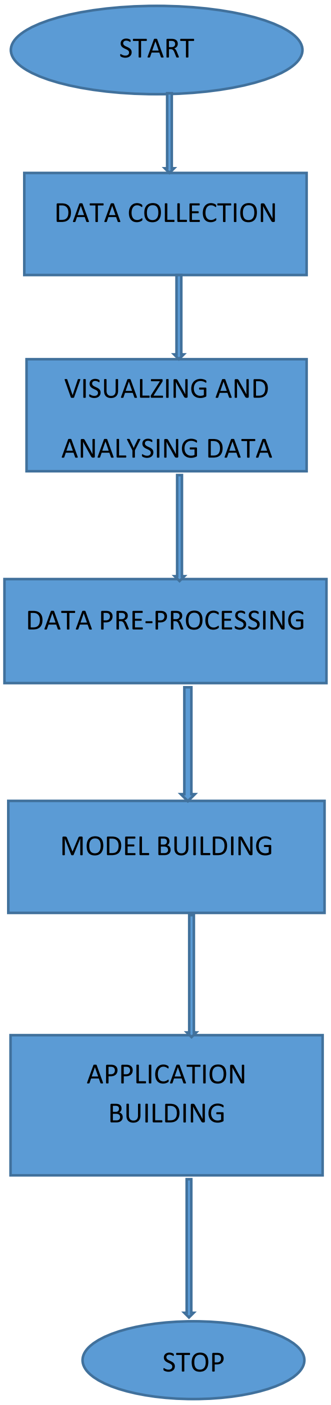
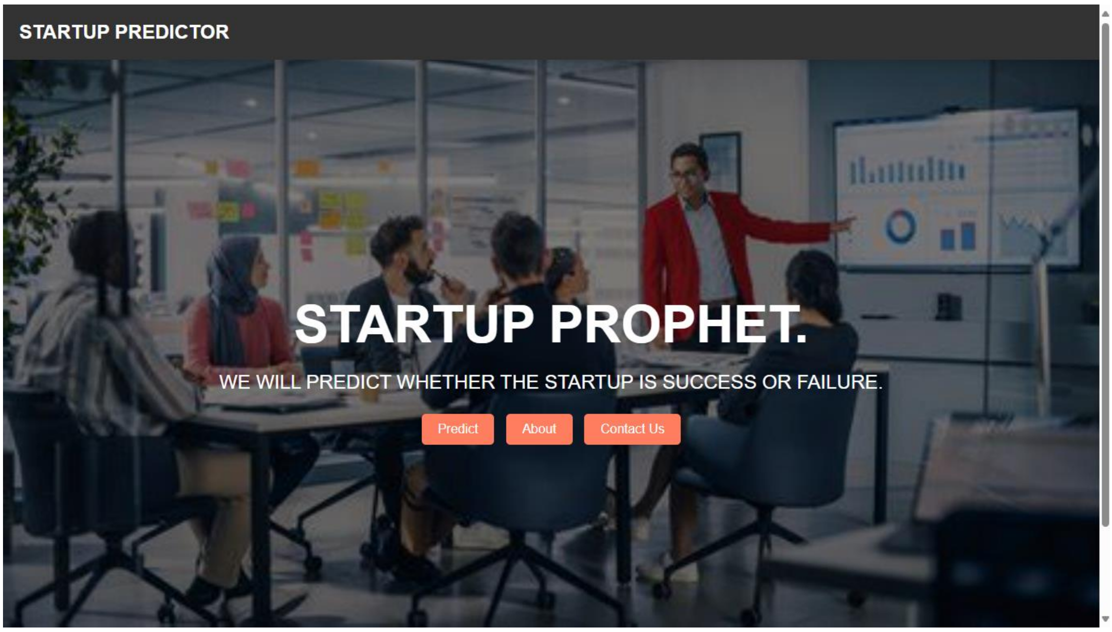


Fig 02: Flow chart of the project

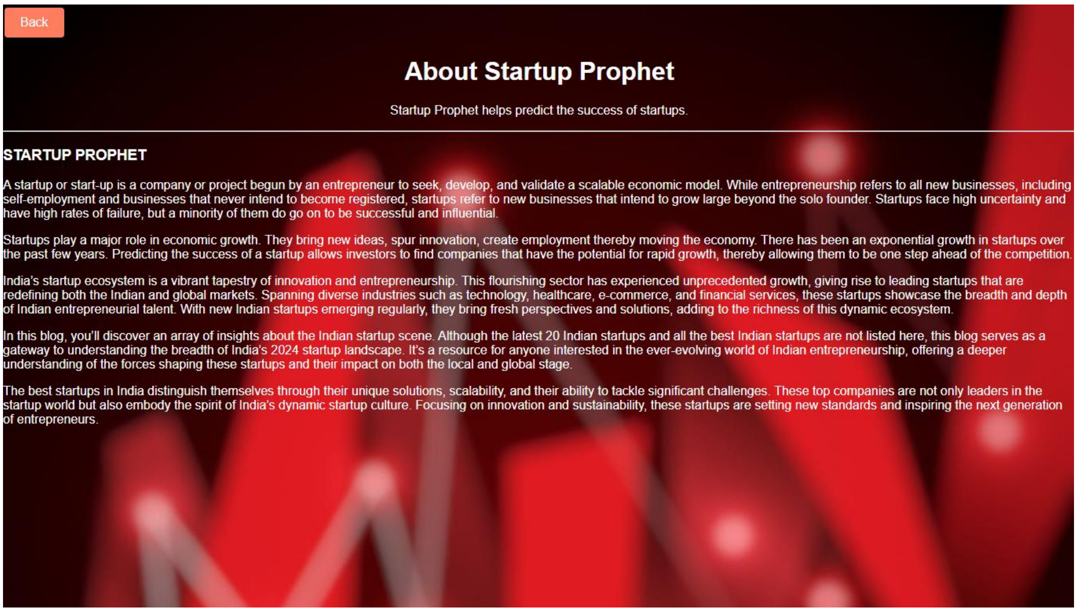
CHAPTER 6

RESULT

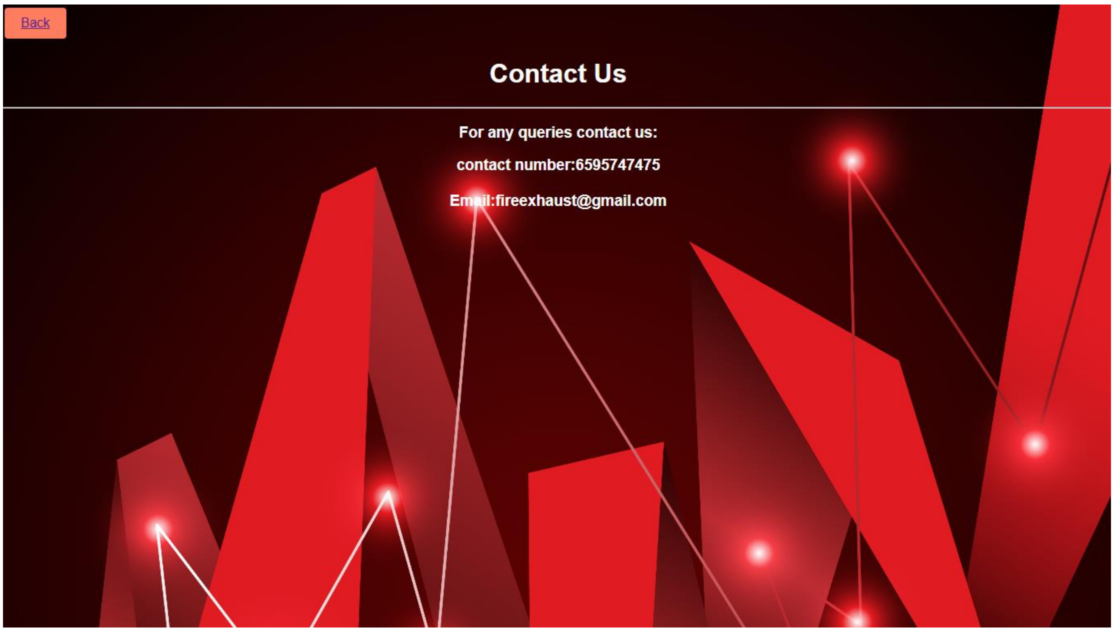
The home page will be shown as:



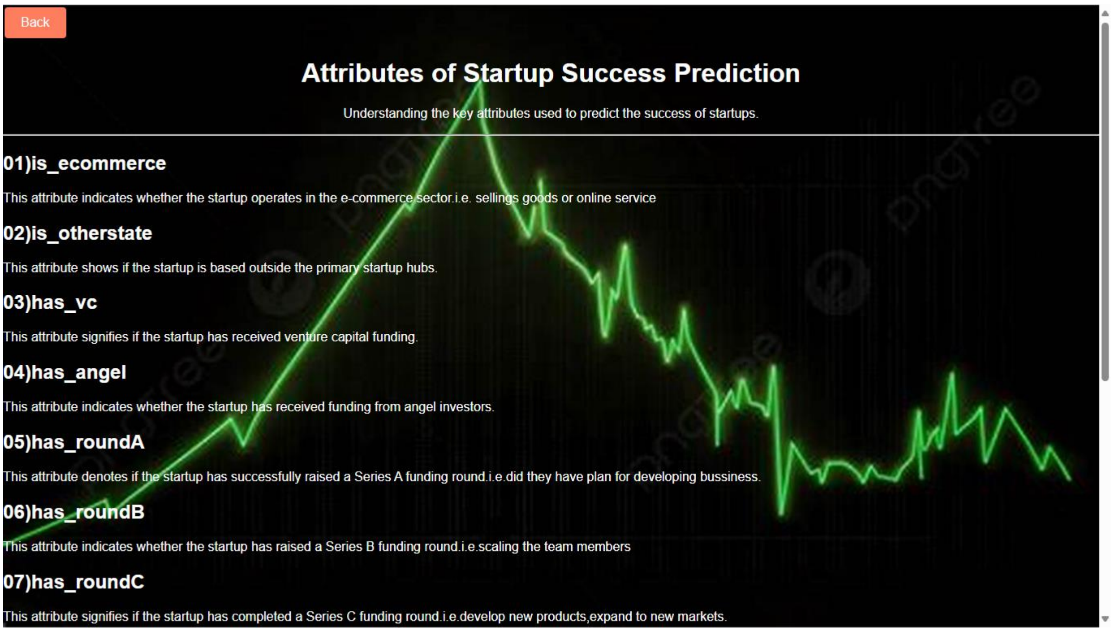
The about page about startup prophet



The contact us page for startup prophet



The info page for startup prophet



The prediction page for startup prophet

The result page shown as

The result page for startup prophet

CHAPTER 7
ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Being your own boss
- Flexibility
- Financial rewards
- Opportunity to innovate
- Chance to impact your community

DISADVANTAGES:

- High costs and limited revenue
- Lack of a developed business model
- Inadequate capital to move to the next phase
- Risk of failure is high
- Long working hours are the norm

CHAPTER 8

APPLICATIONS

The areas where this solution can be used:

- ➔ By stakeholders to predict the growth of the company.
- ➔ By businessman to know whether the company will
  Is Success or failure.

CHAPTER 9

CONCLUSION

FROM THIS PROJECT WE HAVE CONCLUDED THAT:

→ We have Known fundamental concepts and techniques used for machine learning.

→ Gain a broad understanding about data.

→ Have knowledge on pre-processing the data/transformation techniques and some visualization techniques.

CHAPTER 10

FUTURE SCOPE

Enhancements that can be made in the future

→ Know fundamental concepts and techniques used for machine learning.

→ Gain a broad understanding about data.

→ Have knowledge on pre-processing the data/transformation techniques and some visualization concepts.

CHAPTER 11

BIBILOGRAPHY

References of previous works or websites visited/books referred for analysis about the project, previous solution findings and previous STARTUP PROPHET documents.

CHAPTER 12

CODE SNIPPETS

MODEL BUILDING:

1)Dataset
2)Jupyter notebook and Spyder Application Building

1. HTML file (home file, about file, predict file, submit file)
1. Models in pickle format

CODE SNIPPETS

MODEL BUILDING

```
import pandas as pd
import numpy as np
#visulization
from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set(style="whitegrid")
# data preprocessing
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder,StandardScaler,LabelEncoder
from sklearn.preprocessing import StandardScaler,MinMaxScaler
# handling class imbalance
from imblearn.over_sampling import SMOTE
# model
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
# evalution
from sklearn.metrics import accuracy_score,classification_report, recall_score, precision_score, confusion_matrix
import pickle

import warnings
warnings.filterwarnings('ignore')
```

```
In [3]: df=pd.read_csv('startup data.csv')
```

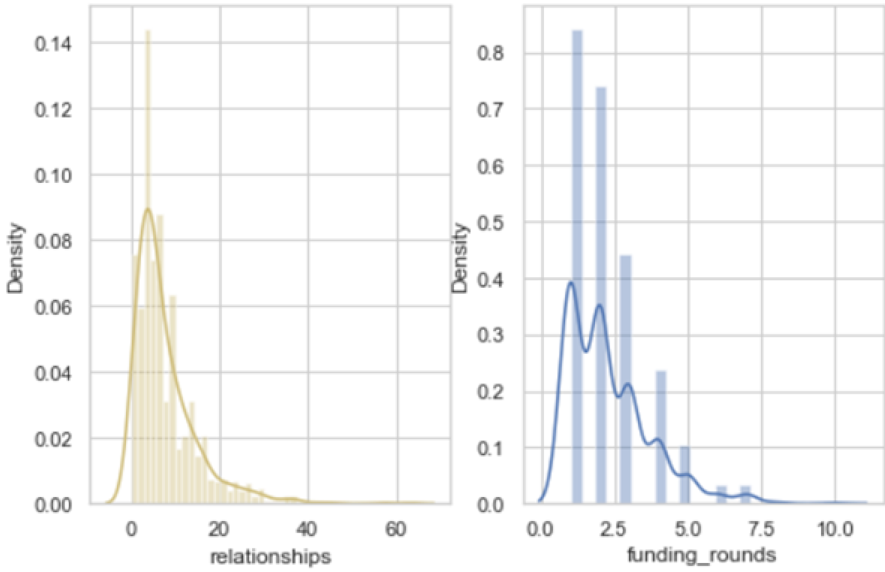
```
In [4]: df
```

Out[4]:

	Unnamed: 0	state_code	latitude	longitude	zip_code	id	city	Unnamed: 6	name	labels	...	object_id	has_VC	has_angel	has_round
0	1005	CA	42.358880	-71.056820	92101	c:6669	San Diego	NaN	Bandsintown	1	...	c:6669	0	1	
1	204	CA	37.238916	-121.973718	95032	c:16283	Los Gatos	NaN	TriCipher	1	...	c:16283	1	0	
2	1001	CA	32.901049	-117.192656	92121	c:65620	San Diego	San Diego CA 92121	Plixi	1	...	c:65620	0	0	
3	738	CA	37.320309	-122.050040	95014	c:42668	Cupertino	Cupertino CA 95014	Solidcore Systems	1	...	c:42668	0	0	
4	1002	CA	37.779281	-122.419236	94105	c:65806	San Francisco	San Francisco CA 94105	Inhale Digital	0	...	c:65806	1	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
918	352	CA	37.740594	-122.376471	94107	c:21343	San Francisco	NaN	CoTweet	1	...	c:21343	0	0	
919	721	MA	42.504817	-71.195611	1803	c:41747	Burlington	Burlington MA 1803	Reef Point Systems	0	...	c:41747	1	0	
920	557	CA	37.408261	-122.015920	94089	c:31549	Sunnyvale	NaN	Paracor Medical	0	...	c:31549	0	0	
921	589	CA	37.556732	-122.288378	94404	c:33198	San Francisco	NaN	Causata	1	...	c:33198	0	0	
922	462	CA	37.386778	-121.966277	95054	c:26702	Santa Clara	Santa Clara CA 95054	Asempra Technologies	1	...	c:26702	0	0	

923 rows × 49 columns

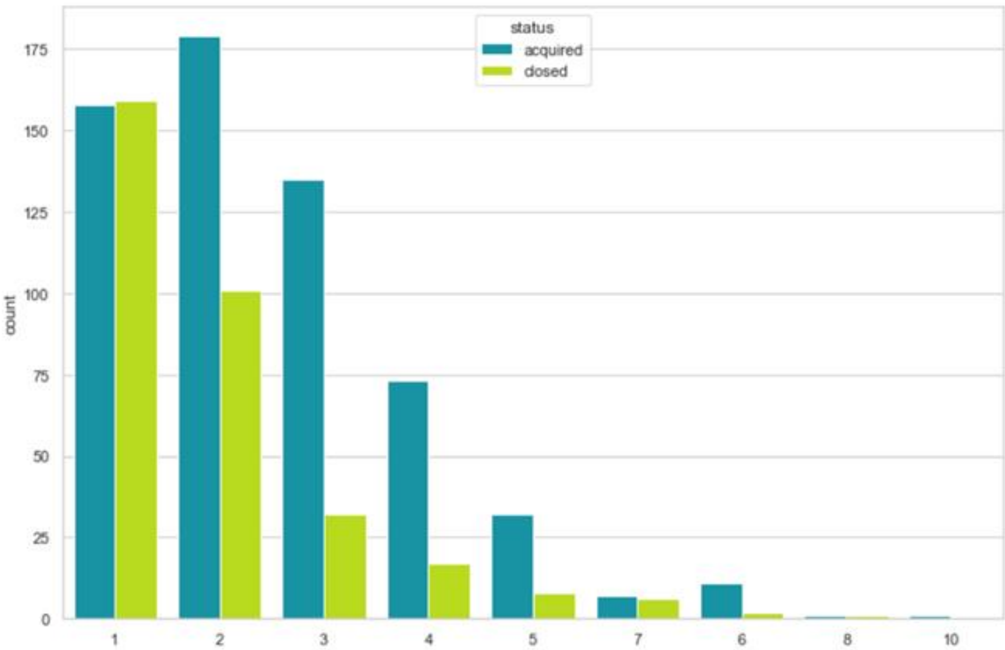
6]:
```
plt.figure(figsize=(12,5))
plt.subplot(131)
sns.distplot(df["relationships"],color="y")
plt.subplot(132)
sns.distplot(df["funding_rounds"])
plt.show()
```



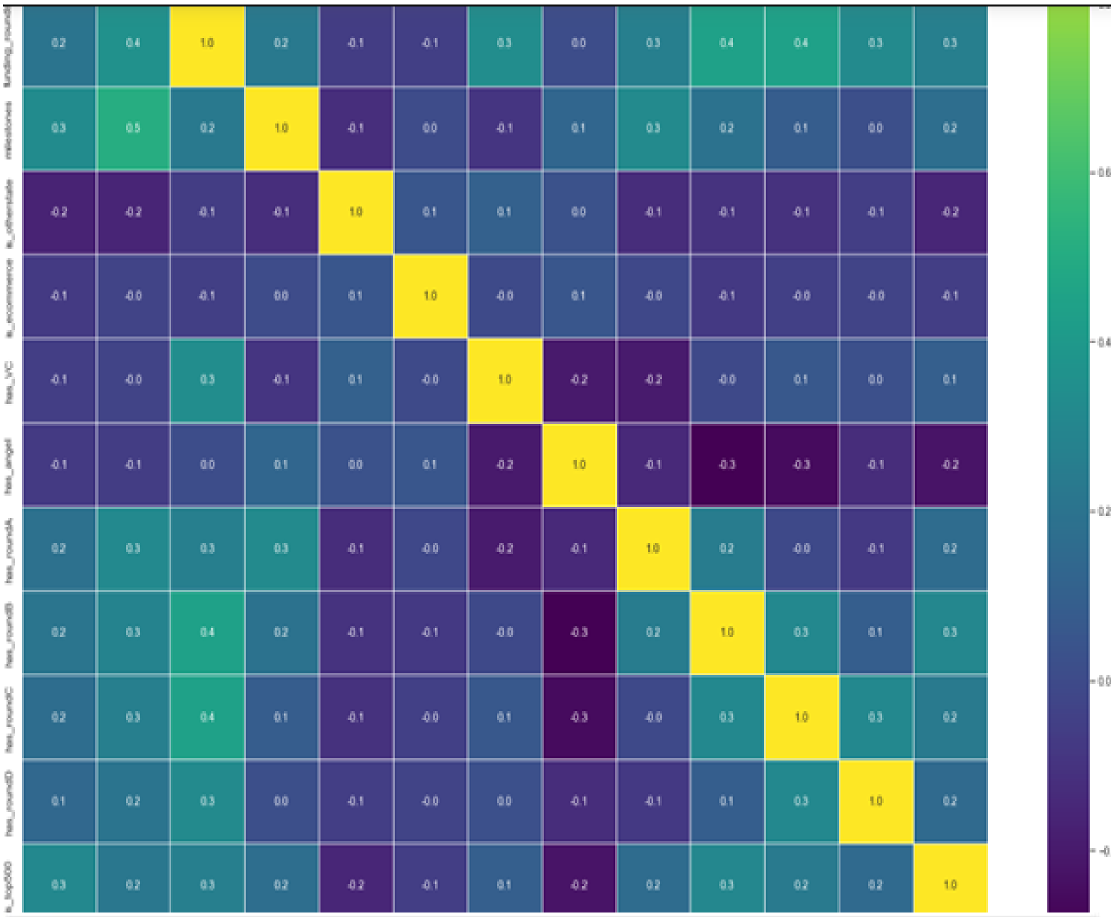
']:
```
fig, ax = plt.subplots(figsize=(12,8))

sns.countplot(x="funding_rounds", hue="status", data=df, palette="nipy_spectral",
              order=df.funding_rounds.value_counts().index)
# plt.legend(bbox_to_anchor=(0.945, 0.90))
```

']:
```
<AxesSubplot:xlabel='funding_rounds', ylabel='count'>
```



]:
```
plt.figure(figsize = (25, 18))
sns.heatmap(df.corr(), annot = True, cmap = 'viridis', linewidth = 0.5, fmt = '.1f')
```



```
df.describe()
```

	Unnamed: 0	latitude	longitude	labels	age_first_funding_year	age_last_funding_year	age_first_milestone_year	age_last_milestone_year	relati
count	923.000000	923.000000	923.000000	923.000000	923.000000	923.000000	771.000000	771.000000	923
mean	572.297941	38.517442	-103.539212	0.646804	2.235630	3.931456	3.055353	4.754423	7
std	333.585431	3.741497	22.394167	0.478222	2.510449	2.967910	2.977057	3.212107	7
min	1.000000	25.752358	-122.756956	0.000000	-9.046600	-9.046600	-14.169900	-7.005500	0
25%	283.500000	37.388869	-122.198732	0.000000	0.576700	1.669850	1.000000	2.411000	3
50%	577.000000	37.779281	-118.374037	1.000000	1.446600	3.528800	2.520500	4.476700	5
75%	866.500000	40.730646	-77.214731	1.000000	3.575350	5.560250	4.686300	6.753400	10
max	1153.000000	59.335232	18.057121	1.000000	21.895900	21.895900	24.684900	24.684900	63

8 rows × 35 columns

5]:
```
df.shape
```
5]: (923, 49)

6]:
```
df.info()
```
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 923 entries, 0 to 922
Data columns (total 49 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Unnamed: 0                923 non-null    int64  
 1   state_code                923 non-null    object 
 2   latitude                  923 non-null    float64
 3   longitude                 923 non-null    float64
 4   zip_code                  923 non-null    object 
 5   id                        923 non-null    object 
 6   city                      923 non-null    object 
 7   Unnamed: 6                430 non-null    object 
 8   name                      923 non-null    object 
 9   labels                    923 non-null    int64  
 10  founded_at                923 non-null    object 
 11  closed_at                 335 non-null    object 
 12  first_funding_at          923 non-null    object 
 13  last_funding_at           923 non-null    object 
 14  age_first_funding_year    923 non-null    float64
 15  age_last_funding_year     923 non-null    float64
 16  age_first_milestone_year  771 non-null    float64
 17  age_last_milestone_year   771 non-null    float64
 18  relationships             923 non-null    int64  
 19  funding_rounds            923 non-null    int64  
 20  funding_total_usd         923 non-null    int64  
 21  milestones                923 non-null    int64  
 22  state_code.1              922 non-null    object 
 23  is_CA                     923 non-null    int64  
 24  is_NY                     923 non-null    int64  
```
S

```
8]: df.isna().sum()

8]: Unnamed: 0                   0
    state_code                   0
    latitude                     0
    longitude                    0
    zip_code                     0
    id                           0
    city                         0
    Unnamed: 6                 493
    name                         0
    labels                       0
    founded_at                   0
    closed_at                  588
    first_funding_at             0
    last_funding_at              0
    age_first_funding_year       0
    age_last_funding_year        0
    age_first_milestone_year   152
    age_last_milestone_year    152
    relationships                0
    funding_rounds               0
    funding_total_usd            0
    milestones                   0
    state_code.1                 1
    is_CA                        0
    is_NY                        0
    is_MA                        0
    is_TX                        0
    is_otherstate                0
    category_code                0
    is_software                  0
    is_web                       0
    is_mobile                    0
    is_enterprise                0
    is_advertising               0
```

```
0]: df.drop(["status","is_othercategory","is_biotech","is_consulting","is_gamesvideo","is_advertising","is_web","is_TX","funding_tota
```

```
df.drop(["is_enterprise","is_CA","is_NY","is_MA","age_last_funding_year","age_first_funding_year","state_code.1","city","last_fur
```

```
[28]: df.shape

[28]: (923, 13)
```

30]: df.dtypes

```
30]: labels            int64
     relationships     int64
     funding_rounds    int64
     milestones        int64
     is_otherstate     int64
     is_ecommerce      int64
     has_VC            int64
     has_angel         int64
     has_roundA        int64
     has_roundB        int64
     has_roundC        int64
     has_roundD        int64
     is_top500         int64
     dtype: object
```

```
In [32]: X = df.drop(['labels'], axis = 1)
         y = df['labels']
```

```
In [33]: sc= StandardScaler()
         x = sc.fit_transform(X)
```

```
In [34]: x
```

```
Out[34]: array([[-0.648696  ,  0.49566485,  0.87613768, ..., -0.55106471,
                 -0.3327311 , -2.06017431],
                [ 0.17754099,  1.21500235, -0.6368185 , ...,  1.81466891,
                  3.00542987,  0.48539582],
                [-0.37328367, -0.94301016,  0.11965959, ..., -0.55106471,
                 -0.3327311 ,  0.48539582],
                ...,
                [-0.37328367, -0.94301016, -0.6368185 , ..., -0.55106471,
                  3.00542987,  0.48539582],
                [ 0.59065949, -0.22367266,  0.11965959, ..., -0.55106471,
                 -0.3327311 ,  0.48539582],
                [-0.51098983, -0.94301016, -0.6368185 , ..., -0.55106471,
                 -0.3327311 ,  0.48539582]])
```

```
In [35]: x = pd.DataFrame(X)
```

```
In [36]: # Apply SMOTE to balance the classes
         smote = SMOTE(sampling_strategy='auto', random_state=42)

         x_bal, y_bal = smote.fit_resample(x, y)
```

```
# Split the dataset into train and test sets
X_train, X_test, y_train, y_test = train_test_split(x_bal, y_bal, test_size=0.3, random_state=42)
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,classification_report, recall_score, precision_score, confusion_matrix, f1_sco
lg=LogisticRegression()

log=lg.fit(x_bal, y_bal)

y_pred=lg.predict(X_test)
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test, y_pred))
```

```
[[140  30]
 [ 54 135]]
              precision    recall  f1-score   support

           0       0.72      0.82      0.77       170
           1       0.82      0.71      0.76       189

    accuracy                           0.77       359
   macro avg       0.77      0.77      0.77       359
weighted avg       0.77      0.77      0.77       359
```

```
from sklearn.metrics import log_loss
logloss = log_loss(y_test,y_pred)
logloss
```

```
8.081563245391097
```

```
In [39]: from sklearn.svm import SVC
         # Train an SVM classifier on the resampled data
         svm = SVC(kernel='rbf',C=2.0, random_state=42)
         svm.fit(x_bal, y_bal)

         # Make predictions on the test set
         y_pred =svm.predict(X_test)

         # Print classification report , confusion matrix
         print(confusion_matrix(y_test,y_pred))
         print(classification_report(y_test, y_pred))
```

```
[[127  43]
 [ 32 157]]
              precision    recall  f1-score   support

           0       0.80      0.75      0.77       170
           1       0.79      0.83      0.81       189

    accuracy                           0.79       359
   macro avg       0.79      0.79      0.79       359
weighted avg       0.79      0.79      0.79       359
```

```
In [40]: from sklearn.ensemble import RandomForestClassifier
         rf = RandomForestClassifier()
         rf.fit(x_bal, y_bal)
         rftrain =rf.predict(X_train)
         rftest =rf.predict(X_test)
         # Print classification report , confusion matrix
         print(confusion_matrix(rftrain,y_train))
         print(confusion_matrix(rftest,y_test))
         print(classification_report(rftrain,y_train))
         print(classification_report(rftest,y_test))

[[409  19]
 [ 18 389]]
[[165   9]
 [  5 180]]
              precision    recall  f1-score   support

           0       0.96      0.96      0.96       428
           1       0.95      0.96      0.95       407

    accuracy                           0.96       835
   macro avg       0.96      0.96      0.96       835
weighted avg       0.96      0.96      0.96       835

              precision    recall  f1-score   support

           0       0.97      0.95      0.96       174
           1       0.95      0.97      0.96       185

    accuracy                           0.96       359
   macro avg       0.96      0.96      0.96       359
weighted avg       0.96      0.96      0.96       359
```

```
In [45]: ## Tesing

         rf.predict([[3,3,3,0,0,0,1,0,0,0,0,0]])

Out[45]: array([1], dtype=int64)
```

```
In [46]: rf.predict([[2,2,1,0,0,1,1,0,0,0,0,1]])

Out[46]: array([0], dtype=int64)
```

```
In [47]: pickle.dump(rf,open("Randf.pkl","wb"))
```