Project Title	**Project Valuation Analysis**
Tools	Machine Learning
Domain	Data Analyst
Project Difficulties level	intermediate

Dataset : Dataset is available in the given link. You can download it at your convenience.

Click here to download data set

About Dataset

"Unicorn" is a term used in the venture capital industry to describe a privately held startup company with a value of over $1 billion. The term was first popularized by venture capitalist Aileen Lee, founder of Cowboy Ventures, a seed-stage venture capital fund based in Palo Alto, California.

Unicorns can also refer to a recruitment phenomenon within the human resources (HR) sector. HR managers may have high expectations to fill a position, leading them to look for candidates with qualifications that are higher than required for a specific job. In essence, these managers are looking for a unicorn, which leads to a disconnect between their ideal candidate versus who they can hire from the pool of people available.

Here's a detailed guide on how to carry out a Valuation Project using machine learning, including a step-by-step explanation and Python code example:

Project Overview

Objective: To analyze and predict the valuation of companies using various machine learning techniques.

Steps to Follow:

1. **Define the Scope and Objective**:
    - Identify the companies you want to analyze.
    - Define the specific objectives of your analysis (e.g., predicting company valuation based on financial metrics).
2. **Data Collection**:
    - Gather relevant financial data from sources such as financial statements, market data, industry reports, etc.
    - Common data points include revenue, EBITDA, net income, free cash flow, enterprise value, etc.
3. **Data Preparation**:
    - Clean the data to remove any inconsistencies or errors.
    - Combine data from different sources into a single dataset.
    - Use tools like Pandas for data cleaning and preparation.
4. **Exploratory Data Analysis (EDA)**:
    - Perform EDA to understand the data distribution and identify patterns.
    - Use visualization tools like Matplotlib and Seaborn to visualize the data.
5. **Feature Engineering**:

    - Create new features from existing data that might be useful for the machine learning model.
    - Normalize or standardize the data if necessary.
6. **Model Selection**:
    - Choose appropriate machine learning algorithms based on the problem (e.g., linear regression, decision trees, random forest, etc.).
    - Split the data into training and testing sets.
7. **Model Training and Evaluation**:
    - Train the machine learning model on the training set.
    - Evaluate the model's performance on the testing set using appropriate metrics.
8. **Model Tuning and Optimization**:
    - Tune the model's hyperparameters to improve performance.
    - Use techniques like cross-validation to ensure the model is not overfitting.
9. **Deployment**:
    - Deploy the model using tools like Flask or Django for web applications.
    - Use the model to make predictions on new data.

Detailed Python Code Example

Step-by-Step Implementation

1. **Data Collection**:
    - Assume you have a dataset named `valuation_data.csv` with columns like `Company`, `Year`, `Revenue`, `EBITDA`, `Net_Income`, `Free_Cash_Flow`, `Enterprise_Value`, etc.

```
# Import necessary libraries
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Load the dataset
data = pd.read_csv('valuation_data.csv')

# Display the first few rows of the dataset
print(data.head())
```

2. **Data Preparation**:

```
# Handle missing values
data = data.dropna()

# Convert categorical columns to numerical (if any)
data = pd.get_dummies(data, drop_first=True)

# Split the data into features and target variable
X = data.drop('Enterprise_Value', axis=1)
y = data['Enterprise_Value']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Standardize the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

3. **Exploratory Data Analysis (EDA)**:

```
# Visualize the distribution of the target variable
sns.histplot(y, kde=True)
plt.title('Distribution of Enterprise Value')
plt.show()

# Visualize correlations between features
plt.figure(figsize=(10, 8))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

4. **Model Selection and Training**:

```
# Initialize the Linear Regression model
lr_model = LinearRegression()

# Train the model on the training data
lr_model.fit(X_train, y_train)

# Make predictions on the testing data
y_pred_lr = lr_model.predict(X_test)

# Initialize the Random Forest Regressor model
rf_model = RandomForestRegressor(random_state=42)

# Train the model on the training data
rf_model.fit(X_train, y_train)

# Make predictions on the testing data
y_pred_rf = rf_model.predict(X_test)
```

5. **Model Evaluation**:

```
# Evaluate the Linear Regression model's performance
```

```
mse_lr = mean_squared_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)

print(f'Linear Regression Mean Squared Error: {mse_lr}')
print(f'Linear Regression R-squared: {r2_lr}')

# Evaluate the Random Forest model's performance
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)

print(f'Random Forest Mean Squared Error: {mse_rf}')
print(f'Random Forest R-squared: {r2_rf}')
```

6. **Model Tuning and Optimization**:

```
from sklearn.model_selection import GridSearchCV

# Define the parameter grid for Random Forest
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10]
}

# Perform Grid Search with cross-validation
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=5,
n_jobs=-1, verbose=2)
grid_search.fit(X_train, y_train)

# Print the best parameters and best score
print(f'Best Parameters: {grid_search.best_params_}')
print(f'Best Score: {grid_search.best_score_}')

# Train the model with the best parameters
best_rf_model = grid_search.best_estimator_
best_rf_model.fit(X_train, y_train)

# Make predictions on the testing data
```

```
y_pred_rf_optimized = best_rf_model.predict(X_test)

# Evaluate the optimized model's performance
mse_rf_optimized = mean_squared_error(y_test, y_pred_rf_optimized)
r2_rf_optimized = r2_score(y_test, y_pred_rf_optimized)

print(f'Optimized Random Forest Mean Squared Error: {mse_rf_optimized}')
print(f'Optimized Random Forest R-squared: {r2_rf_optimized}')
```

7. **Deployment** (Basic Example using Flask):

```
# Import necessary libraries
from flask import Flask, request, jsonify
import joblib

# Initialize the Flask app
app = Flask(__name__)

# Load the trained model
joblib.dump(best_rf_model, 'best_rf_model.pkl')
model = joblib.load('best_rf_model.pkl')

# Define a route for prediction
@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    features = np.array([data['features']])
    prediction = model.predict(features)
    return jsonify({'prediction': prediction[0]})

# Run the app
if __name__ == '__main__':
    app.run(debug=True)
```

Conclusion

By following these steps, you can create a comprehensive company valuation analysis using machine learning. This project will help you understand the valuation process, make data-driven predictions, and deploy a model for real-time predictions.

SAMPLE AND REPORT

Sobre o conjunto de dados

"Unicórnio" é um termo usado na indústria de capital de risco para descrever uma startup de capital fechado com valor superior a US$ 1 bilhão. O termo foi popularizado pela primeira vez pela capitalista de risco Aileen Lee, fundadora da Cowboy Ventures, um fundo de capital de risco com sede em Palo Alto, Califórnia.

Unicórnios também podem se referir a um fenômeno de recrutamento no setor de recursos humanos (RH). Os gerentes de RH podem ter grandes expectativas para preencher um cargo, levando-os a procurar candidatos com qualificações superiores às exigidas para um cargo específico. Em essência, esses gerentes estão procurando um unicórnio, o que leva a uma desconexão entre seu candidato ideal e quem eles podem contratar do grupo de pessoas disponíveis.

Download da base de dados

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

In [2]:

```
base_dados =
pd.read_csv('/kaggle/input/unicorn-startups/unicorns till sep
2022.csv')
```

In [3]:

```
base_dados.shape
```

Out[3]:

```
(1186, 7)
```

In [4]:

```
base_dados.head()
```

Out[4]:

	Company	Valuation ($B)	Date Joined	Country	City	Industry	Investors
0	ByteDance	$140	4/7/2017	China	Beijing	Artificial intelligence	Sequoia Capital China, SIG Asia Investments, S...

1	SpaceX	$127	12/1/2012	United States	Hawthorne	Other	Founders Fund, Draper Fisher Jurvetson, Rothen...
2	SHEIN	$100	7/3/2018	China	Shenzhen	E-commerce & direct-to-consumer	Tiger Global Management, Sequoia Capital China...
3	Stripe	$95	1/23/2014	United States	San Francisco	Fintech	Khosla Ventures, LowercaseCapital, capitalG
4	Canva	$40	1/8/2018	Australia	Surry Hills	Internet software & services	Sequoia Capital China, Blackbird Ventures, Mat...

In [5]:

```
base_dados.columns
```

Out[5]:

```
Index(['Company', 'Valuation ($B)', 'Date Joined', 'Country', 'City ',
       'Industry', 'Investors'],

      dtype='object')
```

In [6]:

```python
# Renomeação
base_dados.rename(columns={'Unnamed: 0' : 'Id', 'Company' :
'Empresa', 'Valuation ($B)': 'Valor ($)', 'Date Joined' : 'Data
de Adesão', 'Country' : 'País',
       'City' : 'Cidade', 'Industry' : 'Setor', 'Select
Investors': 'Investidores'}, inplace=True)
```

In [7]:

```python
base_dados.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1186 entries, 0 to 1185
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Empresa         1186 non-null   object
 1   Valor ($)       1186 non-null   object
 2   Data de Adesão  1186 non-null   object
 3   País            1186 non-null   object
 4   City            1186 non-null   object
 5   Setor           1186 non-null   object
 6   Investors       1168 non-null   object
dtypes: object(7)
memory usage: 65.0+ KB
```

Limpeza de Dados

In [8]:

```python
# Modelando a coluna Valor ($)
```

```
base_dados['Valor ($)'] = pd.to_numeric(base_dados['Valor ($)'].apply(lambda x: x.replace('$', '')))
```

In [9]:

```
# Conversão data
base_dados['Data de Adesão'] = pd.to_datetime(base_dados['Data de Adesão'])
```

In [10]:

```
# criando as colunas mês e ano
base_dados['Mês'] = pd.DatetimeIndex(base_dados['Data de Adesão']).month
base_dados['Ano'] = pd.DatetimeIndex(base_dados['Data de Adesão']).year

base_dados.head()
```

Out[10]:

	Empresa	Valor ($)	Data de Adesão	País	City	Setor	Investors	Mês	Ano
0	ByteDance	140.0	2017-04-07	China	Beijing	Artificial intelligence	Sequoia Capital China, SIG Asia	4	2017

							Investments, S...		
1	SpaceX	127.0	2012-12-01	United States	Hawthorne	Other	Founders Fund, Draper Fisher Jurvetson, Rothen...	12	2012
2	SHEIN	100.0	2018-07-03	China	Shenzhen	E-commerce & direct-to-consumer	Tiger Global Management, Sequoia Capital China...	7	2018
3	Stripe	95.0	2014-01-23	United States	San Francisco	Fintech	Khosla Ventures, LowercaseCapital, capitalG	1	2014
4	Canva	40.0	2018-01-08	Australia	Surry Hills	Internet software & services	Sequoia Capital China, Blackbird Ventures, Mat...	1	2018

In [11]:

```
base_dados[['Investidor_1', 'Investidor_2', 'Investidor_3', 'Investidor_4']] = base_dados['Investors'].str.split(',', expand=True)
```

In [12]:

```
base_dados.drop(columns='Investors', inplace=True)
```

In [13]:

```
base_dados
```

Out[13]:

	Empresa	Valor ($)	Data de Adesão	País	City	Setor	Mês	Ano	Investidor_1	Investidor_2	Investidor_3	Investidor_4
0	ByteDance	140.0	2017-04-07	China	Beijing	Artificial intelligence	4	2017	Sequoia Capital China	SIG Asia Investments	Sina Weibo	Softbank Group
1	SpaceX	127.0	2012-12-01	United States	Hawthorne	Other	12	2012	Founders Fund	Draper Fisher Jurvetson	Rothenberg Ventures	None

2	SHEIN	100.0	2018-07-03	China	Shenzhen	E-commerce & direct-to-consumer	7	2018	Tiger Global Management	Sequoia Capital China	Shunwei Capital Partners	None
3	Stripe	95.0	2014-01-23	United States	San Francisco	Fintech	1	2014	Khosla Ventures	LowercaseCapital	capitalG	None
4	Canva	40.0	2018-01-08	Australia	Surry Hills	Internet software & services	1	2018	Sequoia Capital China	Blackbird Ventures	Matrix Partners	None
...	...	...	...	...	...	...	...	...	...	...	...	...
1181	LeadSquared	1.0	2022-06-21	India	Bengaluru	Internet software & services	6	2022	Gaja Capital Partners	Stakeboat Capital	WestBridge Capital	None

1182	FourKites	1.0	2022-06-21	United States	Chicago	Supply chain, logistics, & delivery	6	2022	Hyde Park Venture Partners	Bain Capital Ventures	Hyde Park Angels	None
1183	VulcanForms	1.0	2022-07-05	United States	Burlington	Supply chain, logistics, & delivery	7	2022	Eclipse Ventures	D1 Capital Partners	Industry Ventures	None
1184	SingleStore	1.0	2022-07-12	United States	San Francisco	Data management & analytics	7	2022	Google Ventures	Accel	Data Collective	None
1185	Unstoppable Domains	1.0	2022-07-27	United States	Las Vegas	Internet software & services	7	2022	Boost VC	Draper Associates	Gaingels	None

1186 rows × 12 columns

In [14]:

```
base_dados.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1186 entries, 0 to 1185
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype         
---  ------           --------------  -----         
 0   Empresa          1186 non-null   object        
 1   Valor ($)        1186 non-null   float64       
 2   Data de Adesão   1186 non-null   datetime64[ns]
 3   País             1186 non-null   object        
 4   City             1186 non-null   object        
 5   Setor            1186 non-null   object        
 6   Mês              1186 non-null   int32         
 7   Ano              1186 non-null   int32         
 8   Investidor_1     1168 non-null   object        
 9   Investidor_2     1118 non-null   object        
 10  Investidor_3     1027 non-null   object        
 11  Investidor_4     10 non-null     object        
dtypes: datetime64[ns](1), float64(1), int32(2), object(8)
memory usage: 102.0+ KB
```

In [15]:

```
# valores nulos
base_dados.isnull().sum()
```

Out[15]:

```
Empresa               0
Valor ($)             0
Data de Adesão        0
País                  0
City                  0
Setor                 0
Mês                   0
Ano                   0
Investidor_1         18
Investidor_2         68
Investidor_3        159
Investidor_4       1176

dtype: int64
```

In [16]:

```
base_dados[base_dados.isnull().T.any()]
```

1176 rows × 12 columns

In [17]:

```
base_dados.isnull().sum()
```

Out[17]:

```
Empresa               0
Valor ($)             0
Data de Adesão        0
País                  0
City                  0
```

```
Setor               0
Mês                 0
Ano                 0
Investidor_1       18
Investidor_2       68
Investidor_3      159
Investidor_4     1176

dtype: int64
```

In [18]:

```
base_dados
```

1186 rows × 12 columns

In [19]:

```
base_dados[base_dados.isnull().T.any()]
```

1176 rows × 12 columns

In [20]:

```
# campos unicos
base_dados.nunique()
```

Out[20]:

```
Empresa          1183
Valor ($)         222
```

```
Data de Adesão     695
País                48
City               286
Setor               34
Mês                 12
Ano                 13
Investidor_1       589
Investidor_2       648
Investidor_3       619
Investidor_4        10

dtype: int64
```

Data visualization

In [21]:

```python
base_dados.groupby('Setor')[['Investidor_1', 'Investidor_2',
'Investidor_3', 'Investidor_4']].count()
```

Out[21]:

	Investidor_1	Investidor_2	Investidor_3	Investidor_4
Setor				
500 Global, Rakuten Ventures, Golden Gate Ventures	0	0	0	0

Andreessen Horowitz, DST Global, IDG Capital	0	0	0	0
Artificial Intelligence	11	11	11	0
Artificial intelligence	74	70	66	2
Auto & transportation	40	38	32	1
B Capital Group, Monk's Hill Ventures, Dynamic Parcel Distribution	0	0	0	0
Consumer & retail	28	25	20	0
Cybersecurity	58	58	56	0
Data management & analytics	45	45	42	1
Dragonfly Captial, Qiming Venture Partners, DST Global	0	0	0	0

E-commerce & direct-to-consumer	103	99	92	0
Edtech	32	31	30	0
Fintech	239	229	215	0
GIC. Apis Partners, Insight Partners	0	0	0	0
Hardware	38	38	34	0
Health	94	90	81	1
Hopu Investment Management, Boyu Capital, DC Thomson Ventures	0	0	0	0
Internet	2	2	2	1
Internet software & services	224	216	200	0

Jungle Ventures, Accel, Venture Highway	0	0	0	0
Kuang-Chi	0	0	0	0
Mobile & telecommunications	36	33	29	2
Mundi Ventures, Doqling Capital Partners, Activant Capital	0	0	0	0
Other	65	55	44	0
Sequoia Capital China, ING, Alibaba Entrepreneurs Fund	0	0	0	0
Sequoia Capital China, Shunwei Capital Partners, Qualgro	0	0	0	0
Sequoia Capital, Thoma Bravo, Softbank	0	0	0	0
SingTel Innov8, Alpha JWC Ventures, Golden Gate Ventures	0	0	0	0

Supply chain, logistics, & delivery	65	64	60	2
Temasek, Guggenheim Investments, Qatar Investment Authority	0	0	0	0
Tiger Global Management, Tiger Brokers, DCM Ventures	0	0	0	0
Travel	14	14	13	0
Vertex Ventures SE Asia, Global Founders Capital, Visa Ventures	0	0	0	0
Vision Plus Capital, GSR Ventures, ZhenFund	0	0	0	0

In [22]:

```
base_dados.loc[base_dados['Setor'] == 'Artificial intelligence', 'Setor'] = 'Artificial Intelligence'
base_dados.loc[base_dados['Setor'] == 'Finttech', 'Setor'] = 'Fintech'
```

In [23]:

```
agrupado = base_dados.groupby('Setor')[['Investidor_1',
'Investidor_2', 'Investidor_3', 'Investidor_4']].count()
agrupado
```

Out[23]:

	Investidor_1	Investidor_2	Investidor_3	Investidor_4
Setor				
500 Global, Rakuten Ventures, Golden Gate Ventures	0	0	0	0
Andreessen Horowitz, DST Global, IDG Capital	0	0	0	0
Artificial Intelligence	85	81	77	2
Auto & transportation	40	38	32	1
B Capital Group, Monk's Hill Ventures,	0	0	0	0

Dynamic Parcel Distribution				
Consumer & retail	28	25	20	0
Cybersecurity	58	58	56	0
Data management & analytics	45	45	42	1
Dragonfly Captial, Qiming Venture Partners, DST Global	0	0	0	0
E-commerce & direct-to-consumer	103	99	92	0
Edtech	32	31	30	0
Fintech	239	229	215	0
GIC. Apis Partners, Insight Partners	0	0	0	0
Hardware	38	38	34	0

Health	94	90	81	1
Hopu Investment Management, Boyu Capital, DC Thomson Ventures	0	0	0	0
Internet	2	2	2	1
Internet software & services	224	216	200	0
Jungle Ventures, Accel, Venture Highway	0	0	0	0
Kuang-Chi	0	0	0	0
Mobile & telecommunications	36	33	29	2
Mundi Ventures, Doqling Capital Partners, Activant Capital	0	0	0	0
Other	65	55	44	0

Sequoia Capital China, ING, Alibaba Entrepreneurs Fund	0	0	0	0
Sequoia Capital China, Shunwei Capital Partners, Qualgro	0	0	0	0
Sequoia Capital, Thoma Bravo, Softbank	0	0	0	0
SingTel Innov8, Alpha JWC Ventures, Golden Gate Ventures	0	0	0	0
Supply chain, logistics, & delivery	65	64	60	2
Temasek, Guggenheim Investments, Qatar Investment Authority	0	0	0	0
Tiger Global Management, Tiger Brokers, DCM Ventures	0	0	0	0
Travel	14	14	13	0

Vertex Ventures SE Asia, Global Founders Capital, Visa Ventures	0	0	0	0
Vision Plus Capital, GSR Ventures, ZhenFund	0	0	0	0

In [24]:

```
agrupado['Total_Investidors'] = agrupado[['Investidor_1',
'Investidor_2', 'Investidor_3', 'Investidor_4']].sum(axis=1)
agrupado.drop(columns=['Investidor_1', 'Investidor_2',
'Investidor_3', 'Investidor_4'], inplace=True)
agrupado1 = agrupado.sort_values(by= 'Total_Investidors',
ascending=False)
agrupado1
```

Out[24]:

	Total_Investidors
Setor	
Fintech	683

Internet software & services	640
E-commerce & direct-to-consumer	294
Health	266
Artificial Intelligence	245
Supply chain, logistics, & delivery	191
Cybersecurity	172
Other	164
Data management & analytics	133
Auto & transportation	111
Hardware	110

Mobile & telecommunications	100
Edtech	93
Consumer & retail	73
Travel	41
Internet	7
Sequoia Capital China, ING, Alibaba Entrepreneurs Fund	0
Tiger Global Management, Tiger Brokers, DCM Ventures	0
Temasek, Guggenheim Investments, Qatar Investment Authority	0
Vertex Ventures SE Asia, Global Founders Capital, Visa Ventures	0

SingTel Innov8, Alpha JWC Ventures, Golden Gate Ventures	0
Sequoia Capital, Thoma Bravo, Softbank	0
Sequoia Capital China, Shunwei Capital Partners, Qualgro	0
500 Global, Rakuten Ventures, Golden Gate Ventures	0
Mundi Ventures, Doqling Capital Partners, Activant Capital	0
Kuang-Chi	0
Jungle Ventures, Accel, Venture Highway	0
Andreessen Horowitz, DST Global, IDG Capital	0

Hopu Investment Management, Boyu Capital, DC Thomson Ventures	0
GIC. Apis Partners, Insight Partners	0
Dragonfly Captial, Qiming Venture Partners, DST Global	0
B Capital Group, Monk's Hill Ventures, Dynamic Parcel Distribution	0
Vision Plus Capital, GSR Ventures, ZhenFund	0

In [25]:

```
# valores unicos do Setor (%)
agrupado =
round(base_dados['Setor'].value_counts(normalize=True) * 100,
2)
agrupado
```

Out[25]:

```
Setor
Fintech
20.15
```

```
Internet software & services
18.89
E-commerce & direct-to-consumer
8.68
Health
7.93
Artificial Intelligence
7.17
Supply chain, logistics, & delivery
5.48
Other
5.48
Cybersecurity
4.89
Data management & analytics
3.79
Auto & transportation
3.37
Hardware
3.20
Mobile & telecommunications
3.12
Edtech
2.70
Consumer & retail
2.36
Travel
1.18
Internet
0.17
Sequoia Capital China, ING, Alibaba Entrepreneurs Fund
0.08
B Capital Group, Monk's Hill Ventures, Dynamic Parcel
Distribution	0.08
Andreessen Horowitz, DST Global, IDG Capital
0.08
Vertex Ventures SE Asia, Global Founders Capital, Visa Ventures
0.08
```

```
Mundi Ventures, Doqling Capital Partners, Activant Capital
0.08
SingTel Innov8, Alpha JWC Ventures, Golden Gate Ventures
0.08
Dragonfly Captial, Qiming Venture Partners, DST Global
0.08
Sequoia Capital China, Shunwei Capital Partners, Qualgro
0.08
Kuang-Chi
0.08
500 Global, Rakuten Ventures, Golden Gate Ventures
0.08
Hopu Investment Management, Boyu Capital, DC Thomson Ventures
0.08
Vision Plus Capital, GSR Ventures, ZhenFund
0.08
GIC. Apis Partners, Insight Partners
0.08
Jungle Ventures, Accel, Venture Highway
0.08
Tiger Global Management, Tiger Brokers, DCM Ventures
0.08
Sequoia Capital, Thoma Bravo, Softbank
0.08
Temasek, Guggenheim Investments, Qatar Investment Authority
0.08

Name: proportion, dtype: float64
```

In [26]:

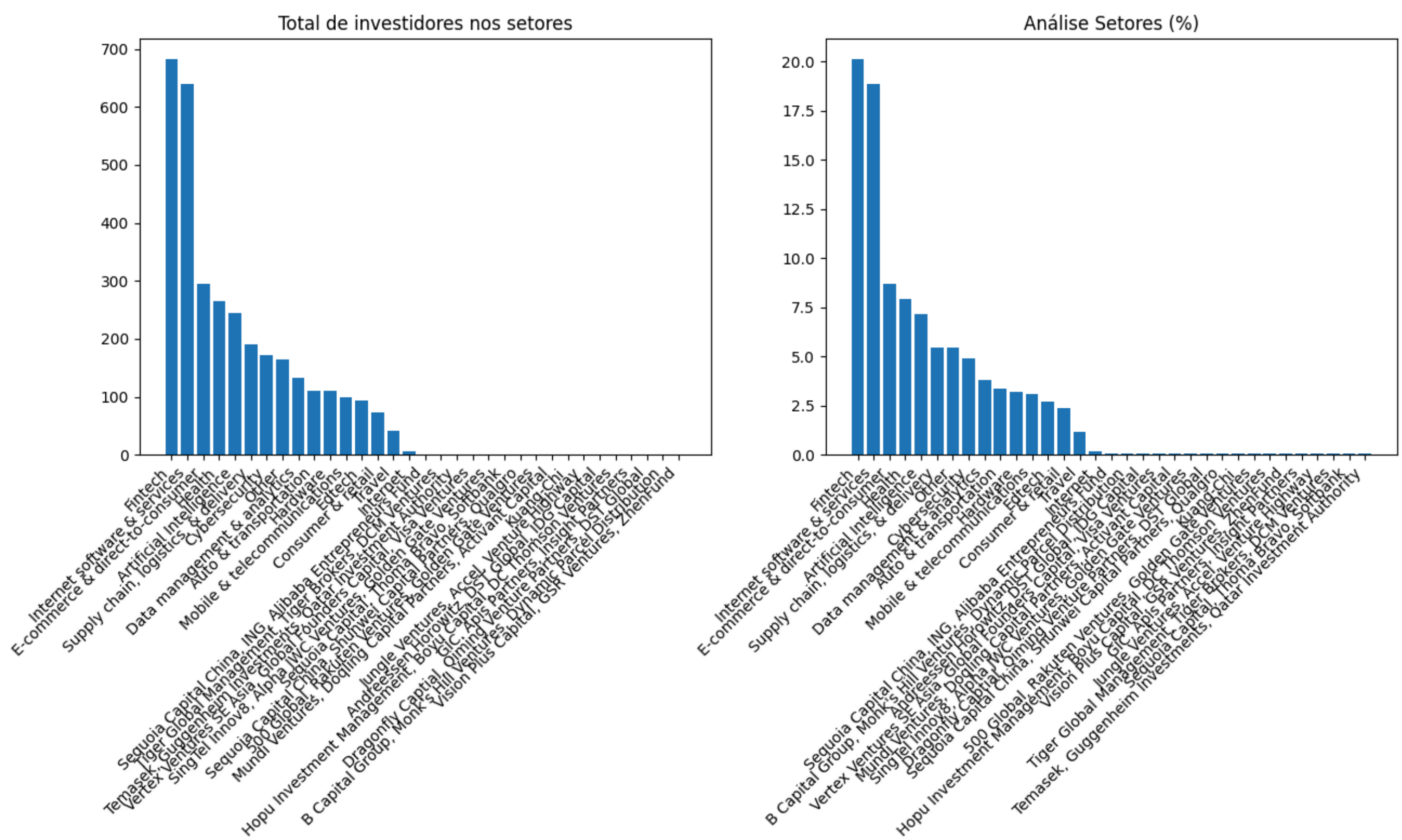
```
plt.figure(figsize=(15,5))

plt.subplot(1,2,1)
plt.title('Total de investidores nos setores')

plt.bar(agrupado1.index, agrupado1.Total_Investidors)
plt.xticks(rotation=45, ha='right')
```

```
plt.subplot(1,2,2)
plt.title('Análise Setores (%)')
plt.bar(agrupado.index, agrupado.values)
plt.xticks(rotation = 45, ha = 'right')

plt.show()
```



In [27]:

```
analise =
round(base_dados['País'].value_counts(normalize=True)*100, 2)
analise = analise.head(10).copy()
```

In [28]:

```
analise
```

Out[28]:

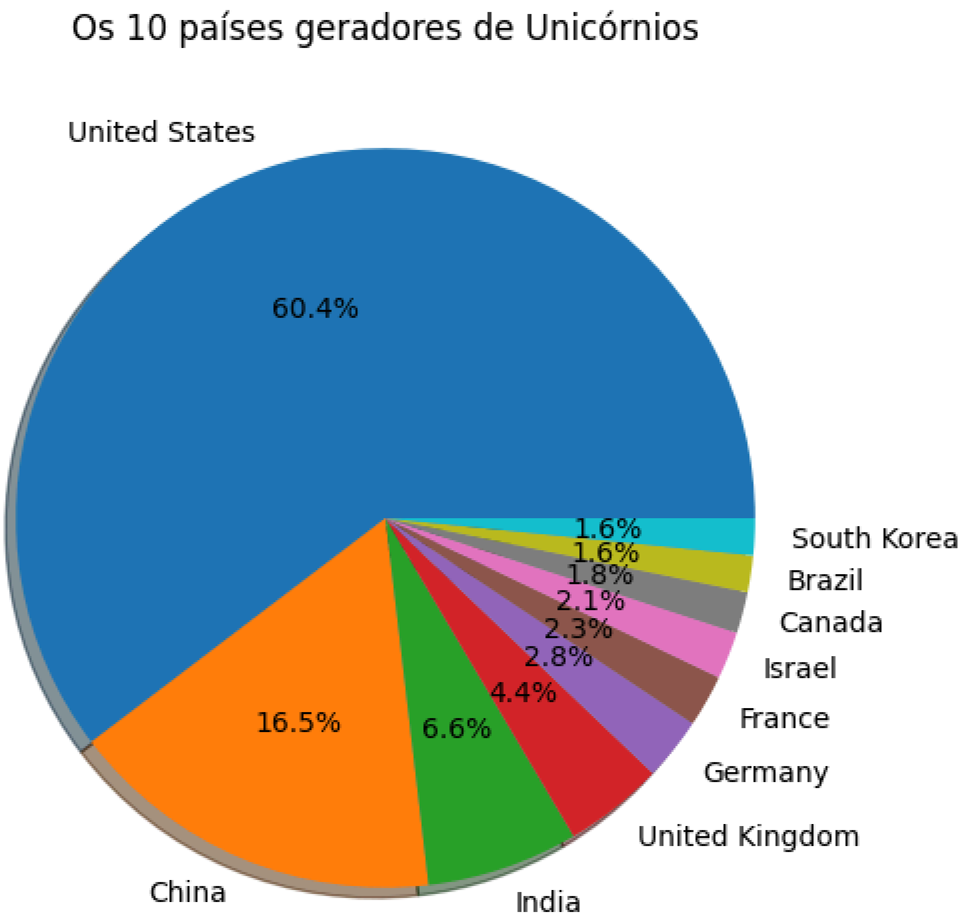
```
País
United States     53.63
China             14.67
India              5.90
United Kingdom     3.88
Germany            2.45
France             2.02
Israel             1.85
Canada             1.60
Brazil             1.43
South Korea        1.43

Name: proportion, dtype: float64
```

In [29]:

```
# Gerando gráfico de pizza dost top 10 países com fintech

plt.figure(figsize=(15,6))
plt.title('Os 10 países geradores de Unicórnios')
plt.pie(
    analise,
    labels=analise.index,
    shadow=True,
    startangle=360,
    autopct='%1.1f%%'
    )
plt.show()
```



In [30]:

```
# Tabela Analítica
agrupamento = base_dados.groupby(by=['País', 'Ano', 'Mês',
'Empresa']).count().reset_index()

agrupamento
```

Out[30]:

	País	An	Mê	Empre	Valor	Data	Cit	Se	Investi	Investi	Investi	Investi

		o	s	sa	($)	de Adesão	y	tor	dor_1	dor_2	dor_3	dor_4
0	Argentina	2021	8	Uala	1	1	1	1	1	1	1	0
1	Australia	2018	1	Canva	1	1	1	1	1	1	1	0
2	Australia	2019	3	Airwallex	1	1	1	1	1	1	1	0
3	Australia	2021	5	Safety Culture	1	1	1	1	1	1	1	0
4	Australia	2021	7	Culture Amp	1	1	1	1	1	1	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...
1181	United State	2022	8	Flow	1	1	1	1	1	0	0	0

	s											
1182	United States	2022	8	Incredible Health	1	1	1	1	1	1	1	0
1183	United States	2022	8	Orna Therapeutics	1	1	1	1	1	1	1	0
1184	Vietnam	2021	10	Sky Mavis	1	1	1	1	1	1	1	0
1185	Vietnam	2021	12	MoMo	1	1	1	1	1	1	1	0

1186 rows × 12 columns

In [31]:

```
agrupamento.loc[agrupamento['País'] == 'Brazil']
```

Out[31]:

	País	Ano	Mês	Empresa	Valor ($)	Data de Adesão	City	Setor	Investidor_1	Investidor_2	Investidor_3	Investidor_4
16	Brazil	2018	7	Movile	1	1	1	1	1	1	1	0
17	Brazil	2018	11	iFood	1	1	1	1	1	1	1	0
18	Brazil	2019	6	Loggi	1	1	1	1	1	1	0	0
19	Brazil	2019	9	QuintoAndar	1	1	1	1	1	1	1	0
20	Brazil	2019	10	EBANX	1	1	1	1	1	1	0	0

21	Brazil	2019	12	Wildlife Studios	1	1	1	1	1	1	0	0
22	Brazil	2020	1	Loft	1	1	1	1	1	1	1	0
23	Brazil	2020	12	C6 Bank	1	1	1	1	1	0	0	0
24	Brazil	2020	12	Creditas	1	1	1	1	1	1	1	0
25	Brazil	2021	1	Madeira Madeira	1	1	1	1	1	1	1	0
26	Brazil	2021	8	Nuvemshop	1	1	1	1	1	1	1	0

27	Brazil	2021	8	Unico	1	1	1	1	1	1	1	0
28	Brazil	2021	9	CloudWalk	1	1	1	1	1	1	1	0
29	Brazil	2021	10	CargoX	1	1	1	1	1	1	1	0
30	Brazil	2021	12	Olist	1	1	1	1	1	1	1	0
31	Brazil	2022	2	Neon	1	1	1	1	1	1	1	0
32	Brazil	2022	5	Dock	1	1	1	1	1	1	1	0

In [32]:

```
agrupamento = base_dados.groupby(by=['País'])['Valor
```

```
($)'].sum().reset_index().sort_values('Valor ($)',
ascending=False)

agrupamento
```

Out[32]:

	País	Valor ($)
46	United States	2069.89
9	China	678.59
45	United Kingdom	205.45
20	India	202.92
18	Germany	80.88

17	France	58.42
1	Australia	54.40
7	Canada	49.23
23	Israel	48.02
6	Brazil	40.08
38	South Korea	34.13
3	Bahamas	32.00
2	Indonesia	29.1

1		3
40	Sweden	23.62
30	Netherlands	22.46
36	Singapore	20.75
19	Hong Kong	20.35
29	Mexico	18.70
43	Turkey	15.77
16	Finland	12.46

41	Switzerland	12.30
22	Ireland	10.05
35	Seychelles	10.00
15	Estonia	9.90
4	Belgium	8.95
25	Japan	8.82
2	Austria	7.61
10	Colombia	7.40

13	Denmark	6.70
39	Spain	6.15
26	Lithuania	6.13
32	Norway	5.70
47	Vietnam	5.27
44	United Arab Emirates	5.05
33	Philippines	3.00
1	Croatia	3.00

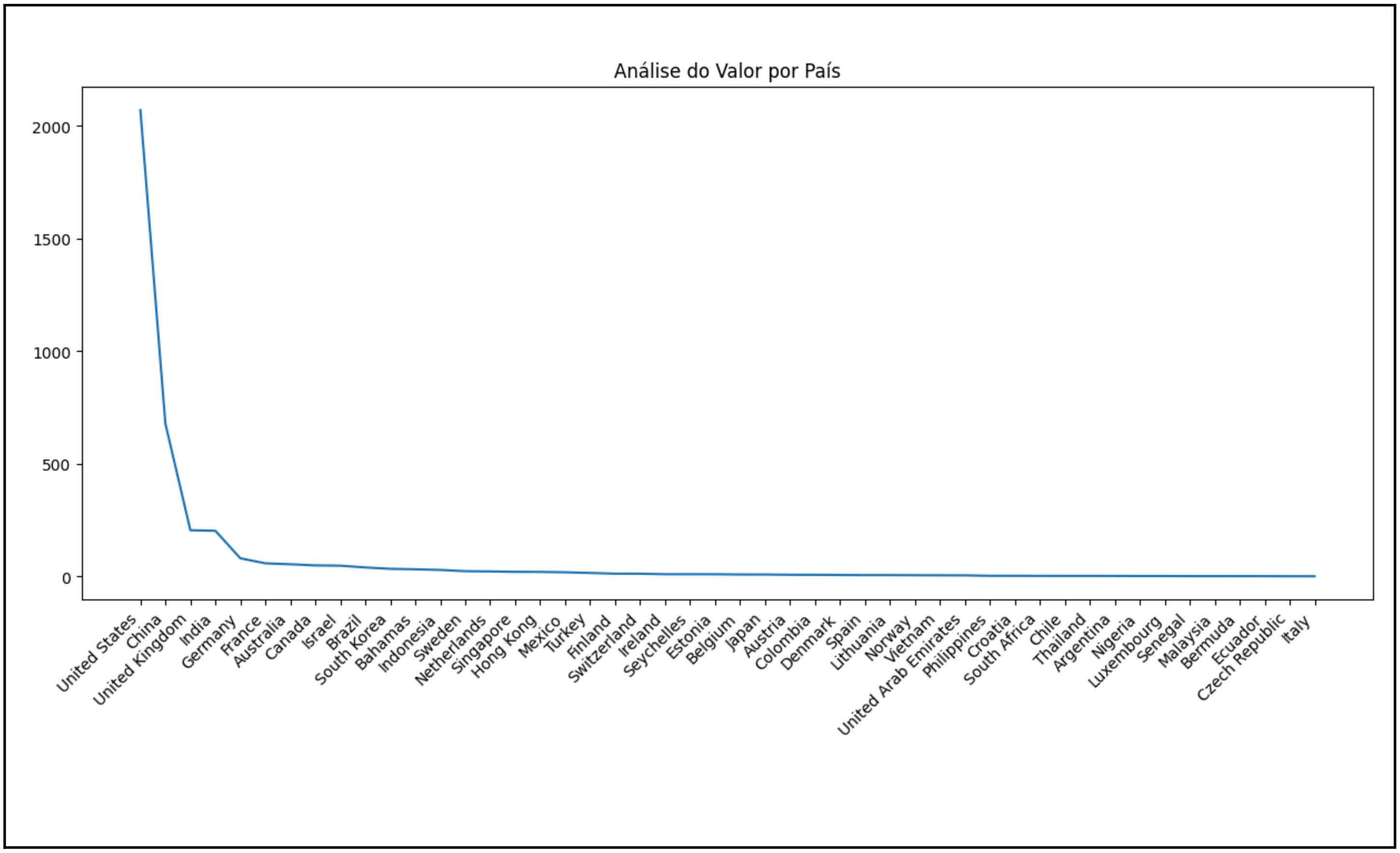
1		
37	South Africa	2.59
8	Chile	2.50
42	Thailand	2.50
0	Argentina	2.45
31	Nigeria	2.00
27	Luxembourg	2.00
34	Senegal	1.70
2	Malaysia	1.70

8		
5	Bermuda	1.60
14	Ecuador	1.50
12	Czech Republic	1.20
24	Italy	1.00

In [33]:

linkcode

```
plt.figure(figsize=(15,6))
plt.title('Análise do Valor por País')
plt.plot(agrupamento['País'], agrupamento['Valor ($)'])
plt.xticks(rotation = 45, ha= 'right')
plt.show()
```



[Reference link]