```
In [1]:
#https://www.youtube.com/watch?v=C64BIMx7Slw
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import sklearn
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import r2_score, classification_report, confusion_matrix
from sklearn import svm
from sklearn.svm import SVR
```

Importing dataset and visualizing dataset

```
In [2]:
#import diabetes dataset
data = pd.read_csv("./Unicorn_Companiesmod.csv" )

#show size of rows and columns
data.shape

#show first 5 rows in dataset
data.head()
```

Out[2]:

	Company	Date Joined	Country	City	Industry	Select Inverstors	Founded Year	Total Raised	Financial Stage	Investors Coun
0	Bytedance	04/07/2017	China	Beijing	Artificial intelligence	Sequoia Capital China, SIG Asia Investments, S...	2012	$7.44B	IPO	28
1	SpaceX	12/01/2012	United States	Hawthorne	Other	Founders Fund, Draper Fisher Jurvetson, Rothen...	2002	$6.874B	None	29
2	Stripe	1/23/2014	United States	San Francisco	Fintech	Khosla Ventures, LowercaseCapital, capitalG	2010	$2.901B	Asset	39
3	Klarna	12/12/2011	Sweden	Stockholm	Fintech	Institutional Venture Partners, Sequoia Capita...	2005	$3.472B	Acquired	56
4	Epic Games	10/26/2018	United States	Cary	Other	Tencent Holdings, KKR, Smash Ventures	1991	$4.377B	Acquired	25

```
In [3]:
#separate the 12 features and the target variable. target variable is outcome column.
X = data.iloc[:, :12]
#show first 5 rows in x
X.head()
```

Out[3]:

	Company	Date Joined	Country	City	Industry	Select Inverstors	Founded Year	Total Raised	Financial Stage	Investors Coun
0	Bytedance	04/07/2017	China	Beijing	Artificial intelligence	Sequoia Capital China, SIG Asia Investments, S...	2012	$7.44B	IPO	28
1	SpaceX	12/01/2012	United States	Hawthorne	Other	Founders Fund, Draper Fisher Jurvetson, Rothen...	2002	$6.874B	None	29
2	Stripe	1/23/2014	United States	San Francisco	Fintech	Khosla Ventures, LowercaseCapital, capitalG	2010	$2.901B	Asset	39
3	Klarna	12/12/2011	Sweden	Stockholm	Fintech	Institutional Venture Partners, Sequoia Capita...	2005	$3.472B	Acquired	56
4	Epic Games	10/26/2018	United States	Cary	Other	Tencent Holdings, KKR, Smash Ventures	1991	$4.377B	Acquired	25

```
In [4]:
#assigning outcome column to y
y = data["Valuation ($B)"]
#show first 5 rows in y
y.head(10)
```

```
Out[4]:
0    140.0
1    100.3
2     95.0
3     45.6
4     42.0
5     40.0
6     40.0
7     39.0
8     38.0
9     33.0
Name: Valuation ($B), dtype: float64
```

```
In [5]:
#visualise data on graph
#sns.pairplot(data,hue='Valuation ($B)',palette='Dark2')
```

Changing sting data to numbers

```
In [6]:
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1037 entries, 0 to 1036
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Company            1037 non-null   object
 1   Date Joined        1037 non-null   object
 2   Country            1037 non-null   object
 3   City               1037 non-null   object
 4   Industry           1037 non-null   object
 5   Select Inverstors  1037 non-null   object
 6   Founded Year       1037 non-null   int64
 7   Total Raised       1037 non-null   object
 8   Financial Stage    1037 non-null   object
 9   Investors Count    1037 non-null   int64
 10  Deal Terms         1037 non-null   int64
 11  Portfolio Exits    1037 non-null   object
 12  Valuation ($B)     1037 non-null   float64
dtypes: float64(1), int64(3), object(9)
memory usage: 105.4+ KB
```

```
In [7]:
data.isnull()
```

Out[7]:

	Company	Date Joined	Country	City	Industry	Select Inverstors	Founded Year	Total Raised	Financial Stage	Investors Count	Deal Terms	Portfo Ex
0	False	False	False	False	False	False	False	False	False	False	False	Fa
1	False	False	False	False	False	False	False	False	False	False	False	Fa
2	False	False	False	False	False	False	False	False	False	False	False	Fa
3	False	False	False	False	False	False	False	False	False	False	False	Fa
4	False	False	False	False	False	False	False	False	False	False	False	Fa
...	...	...	...	...	...	...	...	...	...	...	...	
1032	False	False	False	False	False	False	False	False	False	False	False	Fa
1033	False	False	False	False	False	False	False	False	False	False	False	Fa
1034	False	False	False	False	False	False	False	False	False	False	False	Fa
1035	False	False	False	False	False	False	False	False	False	False	False	Fa
1036	False	False	False	False	False	False	False	False	False	False	False	Fa

1037 rows × 13 columns

```
In [8]:
data.isnull().sum()
```

```
Out[8]:
Company              0
Date Joined          0
Country              0
City                 0
Industry             0
Select Inverstors    0
Founded Year         0
Total Raised         0
Financial Stage      0
Investors Count      0
Deal Terms           0
Portfolio Exits      0
Valuation ($B)       0
dtype: int64
```

```
In [9]:
finStage = pd.get_dummies(data["Financial Stage"])
finStage.head(5)
```

Out[9]:

	Acq	Acquired	Asset	Corporate	Divestiture	IPO	Management	None	Reverse	Take
0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	0

```
In [10]:
indust = pd.get_dummies(data["Industry"])
indust.head(5)
```

Out[10]:

	500 Global, Rakuten Ventures, Golden Gate Ventures	Andreessen Horowitz, DST Global, IDG Capital	Artificial Intelligence	Artificial intelligence	Auto & transportation	B Capital Group, Monk's Hill Ventures, Dynamic Parcel Distribution	Consumer & retail	Cybersecurity	manag & ana
0	0	0	0	1	0	0	0	0	
1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	

5 rows × 33 columns

```
In [11]:
cntry = pd.get_dummies(data["Country"])
cntry.head(5)
```

Out[11]:

	Argentina	Australia	Austria	Bahamas	Belgium	Bermuda	Brazil	Canada	Chile	China	...	South Korea	Spain	Sw
0	0	0	0	0	0	0	0	0	0	1	...	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	

5 rows × 46 columns

```
In [12]:
data.drop("City", axis=1, inplace=True)
```

```
In [13]:
data.drop("Select Inverstors", axis=1, inplace=True)
```

```
In [14]:
invest = pd.get_dummies(data["Portfolio Exits"])
```

```
In [15]:
invest.head(5)
```

Out[15]:

	1	2	3	5	None
0	0	0	0	1	0
1	0	0	0	0	1
2	1	0	0	0	0
3	1	0	0	0	0
4	0	1	0	0	0

```
In [16]:
comp = pd.get_dummies(data["Company"])
comp.head(5)
```

Out[16]:

	1047 Games	1KMXC	1Password	4Paradigm	56PINGTAI	58 Daojia	6Sense	ABL Space Systems	AIWAYS	ASAPP	...	goPuff	iC Ne
0	0	0	0	0	0	0	0	0	0	0	...	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	

5 rows × 1035 columns

```
In [17]:
data.drop("Financial Stage", axis=1, inplace=True)
data.drop("Industry", axis=1, inplace=True)
data.drop("Country", axis=1, inplace=True)
data.drop("Portfolio Exits", axis=1, inplace=True)
data.drop("Company", axis=1, inplace=True)
data.drop("Date Joined", axis=1, inplace=True)
data.drop("Total Raised", axis=1, inplace=True)
```

```
In [18]:
data=pd.concat([data,finStage,indust,cntry,comp],axis=1)
```

```
In [19]:
data.isnull().sum()
```

```
Out[19]:
Founded Year       0
Investors Count    0
Deal Terms         0
Valuation ($B)     0
Acq                0
                  ..
iTutorGroup        0
o9 Solutions       0
reddit             0
solarisBank        0
wefox              0
Length: 1128, dtype: int64
```

```
In [20]:
X=np.array(data['Investors Count']).reshape(-1, 1)
y=np.array(data['Valuation ($B)']).reshape(-1, 1)
X[0:1]
```

```
Out[20]: array([[28]])
```

Splitting the dataset into training and test sets

```
In [21]:
# Split Dataset.
#X = data.drop('Valuation ($B)', axis=1)
#y = data['Valuation ($B)']


#splitting dataset 67% for training and 33% for testing.
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

#show sizes of each variable
print('x train:')
print(X_train.shape)

print('y train:')
print(y_train.shape)

print('x test:')
print(X_test.shape)

print('y test:')
print(y_test.shape)
```

```
x train:
(829, 1)
y train:
(829, 1)
x test:
(208, 1)
y test:
(208, 1)
```

```
In [22]:
#X.head()
```

Scaling dataset

```
In [23]:
# Standard scaling
sc_X = StandardScaler()
X =  pd.DataFrame(sc_X.fit_transform(data.drop(["Valuation ($B)"],axis = 1),))
X.head()
```

Out[23]:

	0	1	2	3	4	5	6	7	8	9	...
0	0.206198	1.365381	2.281538	-0.082439	-0.147224	-0.031068	-0.031068	-0.088173	12.130246	-0.031068	...
1	0.181279	1.465913	4.109235	-0.082439	-0.147224	-0.031068	-0.031068	-0.088173	-0.082439	-0.031068	...
2	0.201214	2.471238	4.109235	-0.082439	-0.147224	32.186954	-0.031068	-0.088173	-0.082439	-0.031068	...
3	0.188755	4.180290	4.566160	-0.082439	6.792375	-0.031068	-0.031068	-0.088173	-0.082439	-0.031068	...
4	0.153868	1.063783	0.910764	-0.082439	6.792375	-0.031068	-0.031068	-0.088173	-0.082439	-0.031068	...

5 rows × 1127 columns

Linear model

```
In [24]:
from sklearn.svm import SVR
svr_linear = SVR(kernel='linear',gamma='scale', C=1.0)
svr_linear.fit(X_train, y_train)
```

```
/opt/jupyterhub/MLenv/lib/python3.8/site-packages/sklearn/utils/validation.py:985: DataConvers
ionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
Out[24]: SVR(kernel='linear')
```

```
In [25]:
svr_linear.score(X_test,y_test)
```

```
Out[25]: -0.0038603935879473195
```

```
In [26]:
y_pred = svr_linear.predict(X_test)

print(r2_score(y_test, y_pred))
```

```
-0.0038603935879473195
```

Poly model

```
In [27]:
svr_poly = SVR(kernel='poly',gamma='scale', C=1.0)
svr_poly.fit(X_train, y_train)
```

```
/opt/jupyterhub/MLenv/lib/python3.8/site-packages/sklearn/utils/validation.py:985: DataConvers
ionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
Out[27]: SVR(kernel='poly')
```

```
In [28]:
svr_poly.score(X_test,y_test)
```

```
Out[28]: -0.02218290618254848
```

```
In [29]:
y_pred = svr_poly.predict(X_test)

print(r2_score(y_test, y_pred))
```

```
-0.02218290618254848
```

RBF model

```
In [30]:
svr_rbf  = SVR(kernel='rbf',gamma='scale', C=1.0)
svr_rbf.fit(X_train, y_train)
```

```
/opt/jupyterhub/MLenv/lib/python3.8/site-packages/sklearn/utils/validation.py:985: DataConvers
ionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
Out[30]: SVR()
```

```
In [31]:
svr_rbf.score(X_test,y_test)
```

```
Out[31]: 0.0005044274059935461
```

```
In [32]:
y_pred = svr_rbf.predict(X_test)

print(r2_score(y_test, y_pred))
```

```
0.0005044274059935461
```