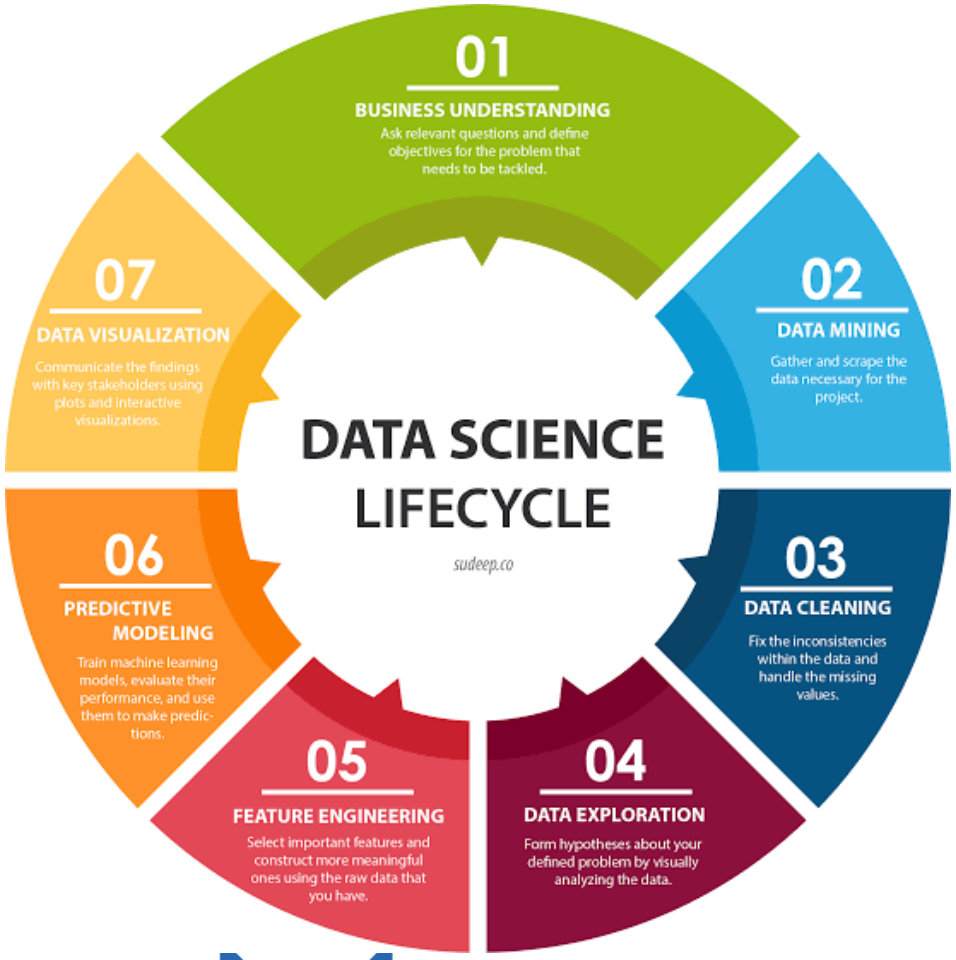
What is Data Science

- Data Science is a multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data.

- Data science is unifying the statistics, data analysis and machine learning



Pandas Library

- Pandas is a software written for python for data manipulation and analysis.
- It offers data structures and operations for manipulating numerical tables.
- Some of its features:
  - Dataframes
  - Reading and writing into the files
  - Column insertion and deletion
  - Data filtration

- import pandas as pd

Data frames

- Pandas is known for its data frames
- A data frame is a two dimensional data structure.
- Data is stored in tubular fashion, as rows and columns.

Id	name	marks
190	Ahmad	89
191	Saeed	80
192	Naeem	92

```python
import pandas as pd

data=[["Ahmad", 20],["Saeed", 24],["Riaz", 23]]

df=pd.DataFrame(data, columns=['Name', 'Marks'])
print (df)
```

```
    Name  Marks
0  Ahmad     20
1  Saeed     24
2   Riaz     23
```

Numpy Library

- NumPy stands for Numerical Python, is a library consisting of multidimensional array objects and routines to process data.

```
1  import numpy as np
2  a=np.array([[1,2,3],[2,4,6]])
3  print(a)
```

```
[[1 2 3]
 [2 4 6]]
```

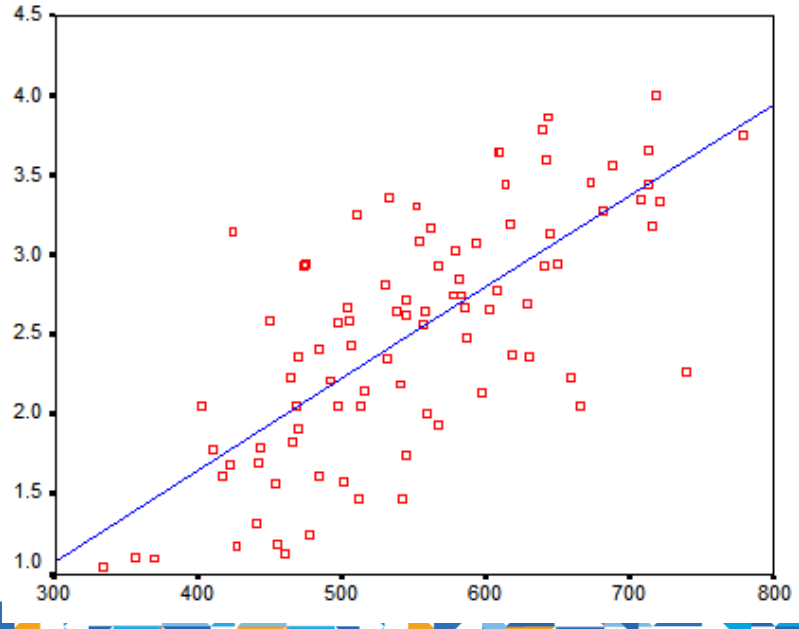
Pre- Requisite

- Pip install scikit-learn / pip3 install scikit-learn
- Afghanistan Population and Data Form Word Bank.
- https://data.worldbank.org/indicator/SP.POP.TOTL?locations=AF
- https://www.kaggle.com/tanuprabhu/population-by-country-2020

Linear Regression

- The Linear regression is a linear approach to modeling the relationships between two variables.
- If there is one explanatory variable it is called Single Linear Regression (SLR)
- SLR is used to estimate the mean line of the different data and predict the future data

- from sklearn.linear_model import LinearRegression



Predicting the Future information

- To predict the future information on any data, we need the following:

- Get the data
- Clean the data (preprocessing)
- Create or Build the model
- Predict the future information
- Plot the data

Import the libraries

```
1 import matplotlib.pyplot as plt
2 from matplotlib.ticker import FormatStrFormatter
3 import pandas as pd
4 import numpy as np
5 from sklearn.linear_model import LinearRegression
6
```

Read the data into Dataframe

```
8  csv_file="population.csv"
9  df=pd.read_csv(csv_file,skiprows=4)
10 print(df)
11
```

1	Data Sour	World Development Indicators													
2															
3	Last Updat	#########													
4															
5	Country N	Country C	Indicator I	Indicator (	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970
6	Aruba	ABW	Populatio	SP.POP.TC	54211	55438	56225	56695	57032	57360	57715	58055	58386	58726	59063
7	Afghanist	AFG	Populatio	SP.POP.TC	8996973	9169410	9351441	9543205	9744781	9956320	10174836	10399926	10637063	10893776	11173642
8	Angola	AGO	Populatio	SP.POP.TC	5454933	5531472	5608539	5679458	5735044	5770570	5781214	5774243	5771652	5803254	5890365
9	Albania	ALB	Populatio	SP.POP.TC	1608800	1659800	1711319	1762621	1814135	1864791	1914573	1965598	2022272	2081695	2135479
10	Andorra	AND	Populatio	SP.POP.TC	13411	14375	15370	16412	17469	18549	19647	20758	21890	23058	24276
11	Arab Worl	ARB	Populatio	SP.POP.TC	92197753	94724510	97334442	1E+08	1.03E+08	1.06E+08	1.09E+08	1.12E+08	1.15E+08	1.18E+08	1.22E+08
12	United Ar	ARE	Populatio	SP.POP.TC	92418	100796	112118	125130	138039	149857	159976	169771	182627	203106	234514
13	Argentina	ARG	Populatio	SP.POP.TC	20481779	20817266	21153052	21488912	21824425	22159650	22494035	22828869	23168267	23517611	23880561
14	Armenia	ARM	Populatio	SP.POP.TC	1874121	1941492	2009526	2077578	2145001	2211319	2276034	2339127	2401143	2462928	2525068
15	American	ASM	Populatio	SP.POP.TC	20123	20602	21253	22034	22854	23672	24462	25248	25989	26703	27363
16	Antigua ar	ATG	Populatio	SP.POP.TC	54131	55001	55841	56702	57641	58698	59915	61241	62521	63550	64177
17	Australia	AUS	Populatio	SP.POP.TC	10276477	10483000	10742000	10950000	11167000	11388000	11651000	11799000	12009000	12263000	12507000

Cleaning the data

- The data should be cleaned to reduce the chances of errors:
- First, print the columns to remove useless ones:

```
1  print (df.columns)
```

```
Index(['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code',
       '1960', '1961', '1962', '1963', '1964', '1965', '1966', '1967', '1968',
       '1969', '1970', '1971', '1972', '1973', '1974', '1975', '1976', '1977',
       '1978', '1979', '1980', '1981', '1982', '1983', '1984', '1985', '1986',
       '1987', '1988', '1989', '1990', '1991', '1992', '1993', '1994', '1995',
       '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
       '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013',
       '2014', '2015', '2016', '2017', '2018', '2019', 'Unnamed: 64'],
      dtype='object')
```

```
1  useless_cols=['Country Code','Indicator Name','Indicator Code', 'Unnamed: 64']
2  df.drop(columns=useless_cols, inplace=True)
3  print(df.columns)
```

Read the specific row of data

```
1  record=df.loc[[1]]
2  print(record)
```

```
  Country Name        1960        1961        196
1  Afghanistan  8996973.0  9169410.0  9351441.

[1 rows x 61 columns]
```

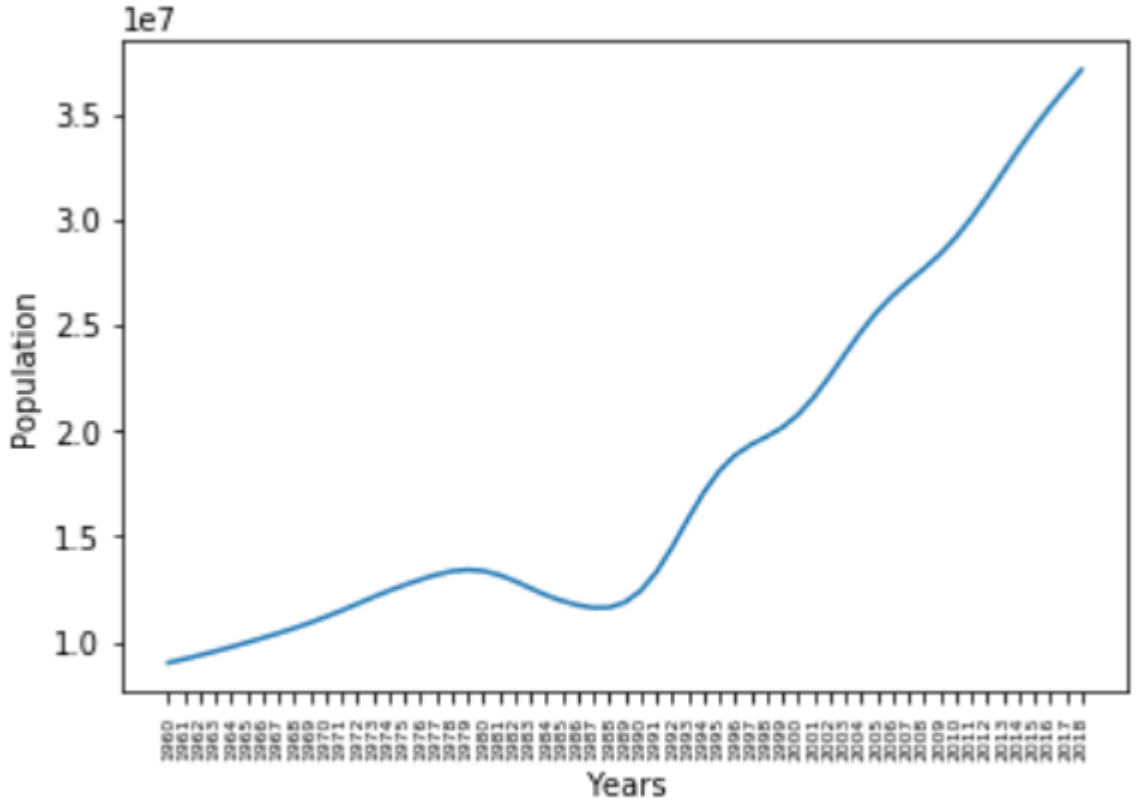
```
4  years=record.columns.tolist()[1:]
5  years=years[0:-1]
6  print(years)
7
8  population=record.values.tolist()[[0][0]]
9  population=population[1:-1]
10 print(population)
```

```
['1960', '1961', '1962', '1963', '1964
9', '1970', '1971', '1972', '1973', '1

[8996973.0, 9169410.0, 9351441.0, 9
10399926.0, 10637063.0, 10893776.0,
```

Plotting the current years

```
1  import matplotlib.pyplot as plt
2
3
4  plt.plot(years,population)
5
6  plt.xlabel("Years")
7  plt.ylabel("Population")
8  plt.xticks(rotation='vertical',fontsize=6)
9
10 plt.show()
11 %matplotlib inline
```



Train the Model and Predict

Train the Model:

```
1  regressor=LinearRegression()
```

```
1  regressor.fit(np.array(years).reshape(-1,1), population)
```

Predict the Future:

```
1  future=round(regressor.predict(np.array([2050]).reshape(-1,1))[0])
2  future
```

```
45547749.0
```

From 2020, 2050

```
future=[]
for years in range(2020,2050):

    predict=round(regressor.predict(np.array([years]).reshape(-1,1))[0])
    future.append(predict)
print(future)

```

```
[32013794.0, 32464926.0, 32916058.0, 33367190.0, 33818321.0, 34269453.0, 347205
85.0, 35171717.0, 35622849.0, 36073981.0, 36525112.0, 36976244.0, 37427376.0, 3
7878508.0, 38329640.0, 38780771.0, 39231903.0, 39683035.0, 40134167.0, 4058529
9.0, 41036431.0, 41487562.0, 41938694.0, 42389826.0, 42840958.0, 43292090.0, 43
743222.0, 44194353.0, 44645485.0, 45096617.0]
```

Add the new data to the previous list

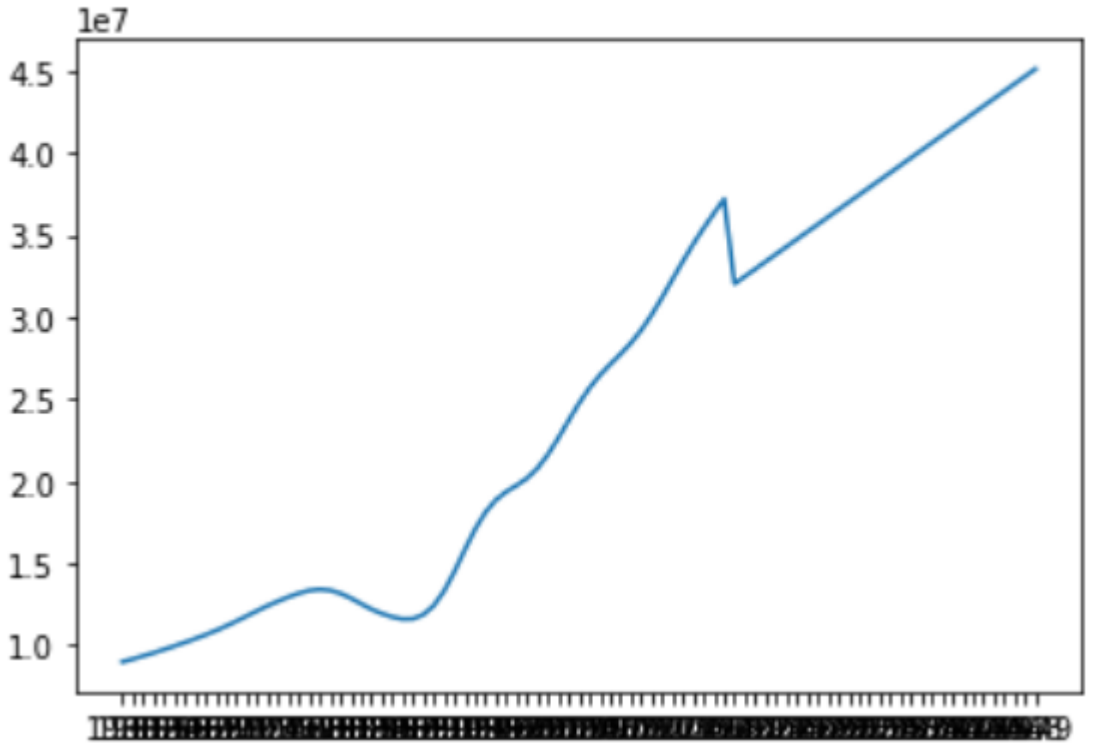
```
for values in future:
    population.append(values)
```

```

for year in range(2020,2050):
    years.append(year)

```

```
plt.plot(years, population)
plt.show()
```



Thank You!