Machine Learning

```
In machine learning there are 4 types of learning:
    1) Supervised learning
    2) Unsupervised learning
    3) Semisupervised learning
    4) Rainforcement learning

Traditional program-
data, coding --> PC --> output

ML-
data, output --> PC --> coding

In supervised learning our data is labeled and it depends on its feature.
In unsupervised learning our data is unlabeled and its depends on cluster.
In semisupervised learning it is combination of supervised $ unsupervised
learning.
In rainforcement learning it is trial and erroe type of data in which our
output shows whenever our data is correct.

*In machine learning if any of our data is qualitative then our model does
not give proper output
so our model is consider as false.
```

Feature Engineering

```
1.Feature tranformation
2.Feature subset selection

There are 6 types of feature tranformation:
    1) MinMaxScaler
    2) Transform Scaler(standard digestion/scaler)
    3) Log Transformation
    4) Power Tranformation
(if we convert quali --> quanti it will gives value error)

    5) One-Hot Encoding
    6) Polynomical Feature
```

MinMaxScaler

In [3]:
```
import pandas as pd
data={
    "number":[1,2,3,4,5],
    "categorical":["A","B","A","B","A"]
}
df=pd.DataFrame(data)
df
```

Out[3]:

	number	categorical
0	1	A
1	2	B
2	3	A
3	4	B
4	5	A

In [14]:
```
import sklearn
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
df['x']=scaler.fit_transform(df[['number']])
df
```

Out[14]:

	number	categorical	x
0	1	A	0.00
1	2	B	0.25
2	3	A	0.50
3	4	B	0.75
4	5	A	1.00

Standard Scaler

In [15]:
```
import sklearn
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
df['x']=scaler.fit_transform(df[['number']])
df
```

Out[15]:

	number	categorical	x
0	1	A	-1.414214
1	2	B	-0.707107
2	3	A	0.000000
3	4	B	0.707107
4	5	A	1.414214

One-Hot Encoding

In [16]:
```
import pandas as pd
data={
    "number":[1,2,3,4,5],
    "categorical":["A","B","A","B","A"]
}
df=pd.DataFrame(data)
df
```

Out[16]:

	number	categorical
0	1	A
1	2	B
2	3	A
3	4	B
4	5	A

In [18]:
```
df1=pd.get_dummies(df)
df1
```

Out[18]:

	number	categorical_A	categorical_B
0	1	1	0
1	2	0	1
2	3	1	0
3	4	0	1
4	5	1	0

In [19]:
```
car=pd.read_csv("car data.csv")
car
```

Out[19]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmi
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	M
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	M
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	M
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	M
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	M
...	...	...	...	...	...	...	...	
296	city	2016	9.50	11.60	33988	Diesel	Dealer	M
297	brio	2015	4.00	5.90	60000	Petrol	Dealer	M
298	city	2009	3.35	11.00	87934	Petrol	Dealer	M
299	city	2017	11.50	12.50	9000	Diesel	Dealer	M
300	brio	2016	5.30	5.90	5464	Petrol	Dealer	M

301 rows × 9 columns

In [23]:
```
car.drop(columns=["Car_Name"],inplace=True)
car
```

Out[23]:

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owne
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	
1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	
...	...	...	...	...	...	...	...	...
296	2016	9.50	11.60	33988	Diesel	Dealer	Manual	
297	2015	4.00	5.90	60000	Petrol	Dealer	Manual	
298	2009	3.35	11.00	87934	Petrol	Dealer	Manual	
299	2017	11.50	12.50	9000	Diesel	Dealer	Manual	
300	2016	5.30	5.90	5464	Petrol	Dealer	Manual	

301 rows × 8 columns

In [24]:
```
car1=pd.get_dummies(car)
car1
```

Out[24]:

	Year	Selling_Price	Present_Price	Kms_Driven	Owner	Fuel_Type_CNG	Fuel_Type_Diesel
0	2014	3.35	5.59	27000	0	0	0
1	2013	4.75	9.54	43000	0	0	1
2	2017	7.25	9.85	6900	0	0	0
3	2011	2.85	4.15	5200	0	0	0
4	2014	4.60	6.87	42450	0	0	1
...	...	...	...	...	...	...	...
296	2016	9.50	11.60	33988	0	0	1
297	2015	4.00	5.90	60000	0	0	0
298	2009	3.35	11.00	87934	0	0	0
299	2017	11.50	12.50	9000	0	0	1
300	2016	5.30	5.90	5464	0	0	0

301 rows × 12 columns

In [33]:
```
df=pd.read_csv("car data.csv")
df
```

Out[33]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmi
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	M
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	M
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	M
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	M
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	M
...	...	...	...	...	...	...	...	
296	city	2016	9.50	11.60	33988	Diesel	Dealer	M
297	brio	2015	4.00	5.90	60000	Petrol	Dealer	M
298	city	2009	3.35	11.00	87934	Petrol	Dealer	M
299	city	2017	11.50	12.50	9000	Diesel	Dealer	M
300	brio	2016	5.30	5.90	5464	Petrol	Dealer	M

301 rows × 9 columns

In [34]:
```
df.drop(columns='Car_Name',inplace=True)
df
```

Out[34]:

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owne
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	
1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	
...	...	...	...	...	...	...	...	
296	2016	9.50	11.60	33988	Diesel	Dealer	Manual	
297	2015	4.00	5.90	60000	Petrol	Dealer	Manual	
298	2009	3.35	11.00	87934	Petrol	Dealer	Manual	
299	2017	11.50	12.50	9000	Diesel	Dealer	Manual	
300	2016	5.30	5.90	5464	Petrol	Dealer	Manual	

301 rows × 8 columns

In [35]:
```
pd.get_dummies(df)
```

Out[35]:

	Year	Selling_Price	Present_Price	Kms_Driven	Owner	Fuel_Type_CNG	Fuel_Type_Diesel
0	2014	3.35	5.59	27000	0	0	0
1	2013	4.75	9.54	43000	0	0	1
2	2017	7.25	9.85	6900	0	0	0
3	2011	2.85	4.15	5200	0	0	0
4	2014	4.60	6.87	42450	0	0	1
...	...	...	...	...	...	...	...
296	2016	9.50	11.60	33988	0	0	1
297	2015	4.00	5.90	60000	0	0	0
298	2009	3.35	11.00	87934	0	0	0
299	2017	11.50	12.50	9000	0	0	1
300	2016	5.30	5.90	5464	0	0	0

301 rows × 12 columns

In [36]:
```
data1=pd.get_dummies(df,drop_first=True)
data1
```

Out[36]:

	Year	Selling_Price	Present_Price	Kms_Driven	Owner	Fuel_Type_Diesel	Fuel_Type_Petrol
0	2014	3.35	5.59	27000	0	0	1
1	2013	4.75	9.54	43000	0	1	0
2	2017	7.25	9.85	6900	0	0	1
3	2011	2.85	4.15	5200	0	0	1
4	2014	4.60	6.87	42450	0	1	0
...	...	...	...	...	...	...	...
296	2016	9.50	11.60	33988	0	1	0
297	2015	4.00	5.90	60000	0	0	1
298	2009	3.35	11.00	87934	0	0	1
299	2017	11.50	12.50	9000	0	1	0
300	2016	5.30	5.90	5464	0	0	1

301 rows × 9 columns

In [3]:
```
import pandas as pd
data={
    "lab":[1,2,3,4,5,6,7],
    "length":[15,14,16,8,8,8,20],
    "width":[30,30,30,30,30,30,30]
}
df=pd.DataFrame(data)
df
```

Out[3]:

	lab	length	width
0	1	15	30
1	2	14	30
2	3	16	30
3	4	8	30
4	5	8	30
5	6	8	30
6	7	20	30

In [4]:
```
# adding Feature
df['area']=df['length']*df['width']
df
```

Out[4]:

	lab	length	width	area
0	1	15	30	450
1	2	14	30	420
2	3	16	30	480
3	4	8	30	240
4	5	8	30	240
5	6	8	30	240
6	7	20	30	600

In [6]:
```
df[df['area']<250]
```

Out[6]:

	lab	length	width	area
3	4	8	30	240
4	5	8	30	240
5	6	8	30	240

In [10]:
```
import numpy as np
# np.where(condition,x,y)
arr=np.array([1,2,3,4,5,6,7,8])
x=np.where(arr%2==1,'odd','even')
print(x)
```

```
['odd' 'even' 'odd' 'even' 'odd' 'even' 'odd' 'even']
```

In [14]:
```
import numpy as np
arr=np.array([1,2,3,4,5,6,7,8,1.5,2.5])
x=np.where(arr%2==1,'odd',np.where(arr%2==0,'even','float'))
print(x)
```

```
['odd' 'even' 'odd' 'even' 'odd' 'even' 'odd' 'even' 'float' 'float']
```

In [15]:
```
import pandas as pd
data={
    "lab":[1,2,3,4,5,6,7],
    "length":[15,14,16,8,8,8,20],
    "width":[30,30,30,30,30,30,30]
}
df=pd.DataFrame(data)
df
```

Out[15]:

	lab	length	width
0	1	15	30
1	2	14	30
2	3	16	30
3	4	8	30
4	5	8	30
5	6	8	30
6	7	20	30

In [16]:
```
# adding Feature
df['area']=df['length']*df['width']
df
```

Out[16]:

	lab	length	width	area
0	1	15	30	450
1	2	14	30	420
2	3	16	30	480
3	4	8	30	240
4	5	8	30	240
5	6	8	30	240
6	7	20	30	600

In [20]:
```
import numpy as np
df['lab_type']=np.where(df['area']<250,"small",np.where(df['area']>=500,"large"
df
```

Out[20]:

	lab	length	width	area	lab_type
0	1	15	30	450	medium
1	2	14	30	420	medium
2	3	16	30	480	medium
3	4	8	30	240	small
4	5	8	30	240	small
5	6	8	30	240	small
6	7	20	30	600	large

In [ ]: