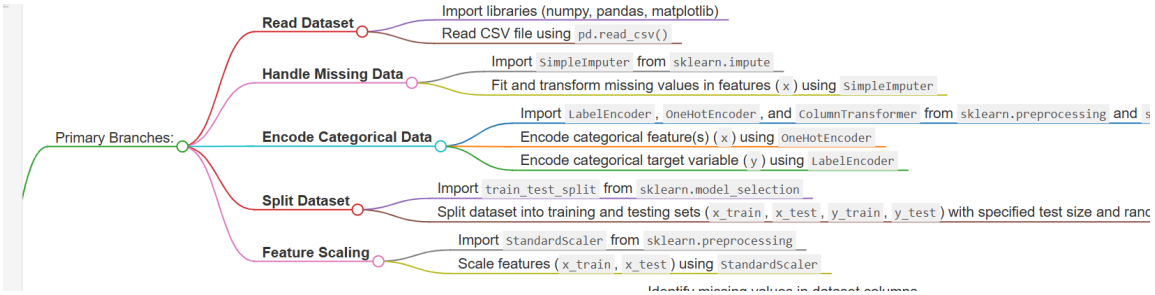
In [1]:
```
import numpy as np
import matplotlib.pyplot as pt
import pandas as pd
```

In [2]:
```
data_set=pd.read_csv(r"C:\Users\LakshmiSrinivas\Desktop\Day wise Assignments  to do
```



In [4]:
```
data_set
```

Out[4]:

	State	Age	Salary	Purchased
0	Mumbai	44.0	72000.0	No
1	Bangalore	27.0	48000.0	Yes
2	Hyderabad	30.0	54000.0	No
3	Bangalore	38.0	61000.0	No
4	Hyderabad	40.0	NaN	Yes
5	Mumbai	35.0	58000.0	Yes
6	Bangalore	NaN	52000.0	No
7	Mumbai	48.0	79000.0	Yes
8	Hyderabad	50.0	83000.0	No
9	Mumbai	37.0	67000.0	Yes

In [11]:
```
x=data_set.iloc[:,:-1].values
```

In [12]:
```
y=data_set.iloc[:,3].values
```

In [8]:
```
x  #indepent variable
```

Out[8]:

	State	Age	Salary
0	Mumbai	44.0	72000.0
1	Bangalore	27.0	48000.0
2	Hyderabad	30.0	54000.0
3	Bangalore	38.0	61000.0
4	Hyderabad	40.0	NaN
5	Mumbai	35.0	58000.0
6	Bangalore	NaN	52000.0
7	Mumbai	48.0	79000.0
8	Hyderabad	50.0	83000.0
9	Mumbai	37.0	67000.0

In [18]:
```
pd.DataFrame(y) # dependent or target variable
```

Out[18]:

	0
0	No
1	Yes
2	No
3	No
4	Yes
5	Yes
6	No
7	Yes
8	No
9	Yes

In [14]:
```
# fill the missing values for x table  using sklearn
from sklearn.impute  import SimpleImputer
imputer=SimpleImputer()
imputer=imputer.fit(x[:,1:3])
x[:,1:3]=imputer.transform(x[:,1:3])
```

In [19]:
```
pd.DataFrame(x)      #here null values are replace with  mean value by default
```

Out[19]:

	0	1	2
0	Mumbai	44.0	72000.0
1	Bangalore	27.0	48000.0
2	Hyderabad	30.0	54000.0
3	Bangalore	38.0	61000.0
4	Hyderabad	40.0	63777.777778
5	Mumbai	35.0	58000.0
6	Bangalore	38.777778	52000.0
7	Mumbai	48.0	79000.0
8	Hyderabad	50.0	83000.0
9	Mumbai	37.0	67000.0

In [22]:
```
#Categorize variable and create dummy variable use the encoder
from sklearn.preprocessing import LabelEncoder
labelencoder_x=LabelEncoder()
labelencoder_x.fit_transform(x[:,0])
x[:,0]=labelencoder_x.fit_transform(x[:,0])
```

In [24]:
```
pd.DataFrame(x)  #Encoder applied to independent variable x (place)  is categorized
```

Out[24]:

	0	1	2
0	2	44.0	72000.0
1	0	27.0	48000.0
2	1	30.0	54000.0
3	0	38.0	61000.0
4	1	40.0	63777.777778
5	2	35.0	58000.0
6	0	38.777778	52000.0
7	2	48.0	79000.0
8	1	50.0	83000.0
9	2	37.0	67000.0

In [25]:
```
labelencoder_y=LabelEncoder()   #Label Encoder to convert categorical to numerical
y=labelencoder_y.fit_transform(y)
```

In [26]:
```
pd.DataFrame(y)  #dependent variable is categorized to numeriacal values
```

Out[26]:

	0
0	0
1	1
2	0
3	0
4	1
5	1
6	0
7	1
8	0
9	1

In [34]:
```
#split the data into training set and  testing set
from sklearn.model_selection   import  train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
#Here Random variable used for accurate behaviour of model
```

In [38]:
```
pd.DataFrame(x_train)      #x-Independent variable is trainied  by 80%
```

Out[38]:

	0	1	2
0	1	40.0	63777.777778
1	2	37.0	67000.0
2	0	27.0	48000.0
3	0	38.777778	52000.0
4	2	48.0	79000.0
5	0	38.0	61000.0
6	2	44.0	72000.0
7	2	35.0	58000.0

In [39]:
```
pd.DataFrame(x_test)  #x-Independent variable is tested by 20%
```

Out[39]:

	0	1	2
0	1	30.0	54000.0
1	1	50.0	83000.0

In [40]:
```
pd.DataFrame(y_train)    #Dependent(target) variable is trained by  80%
```

Out[40]:

	0
0	1
1	1
2	1
3	0
4	1
5	0
6	0
7	1

In [42]:
```
pd.DataFrame(y_test)  #Depedent variable trained by 20%
```

Out[42]:

	0
0	0
1	0

In [49]:
```
# Feautre scaling
from sklearn.preprocessing import Normalizer
sc_x=Normalizer()
x_train=sc_x.fit_transform(x_train)
x_test=sc_x.transform(x_test)
```

In [50]:
```
pd.DataFrame(x_train)   #normalized values  of training data
```

Out[50]:

	0	1	2
0	0.000016	0.000627	1.0
1	0.000030	0.000552	1.0
2	0.000000	0.000562	1.0
3	0.000000	0.000746	1.0
4	0.000025	0.000608	1.0
5	0.000000	0.000623	1.0
6	0.000028	0.000611	1.0
7	0.000034	0.000603	1.0

In [51]:
```
pd.DataFrame(x_test)   #normalized values of testing data
```

Out[51]:

	0	1	2
0	0.000019	0.000556	1.0
1	0.000012	0.000602	1.0

In [ ]: