```
from google.colab import files
upload=files.upload()
```

```
Choose Files SalaryData_Test.csv
• SalaryData_Test.csv(text/csv) - 1694474 bytes, last modified: 2/24/2023 - 100% done
Saving SalaryData_Test.csv to SalaryData_Test.csv
```

```
import pandas as pd
sal_test=pd.read_csv("SalaryData_Test.csv")
sal_test.shape
sal_test.head()
```

	age	workclass	education	educationno	maritalstatus	occupation	relationship	rac
0	25	Private	11th	7	Never-married	Machine-op-inspct	Own-child	Blac
1	38	Private	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	Whit
2	28	Local-gov	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	Whit
3	44	Private	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Blac
4	34	Private	10th	6	Never-married	Other-service	Not-in-family	Whit

```
sal_test.isnull().sum()
sal_test.dtypes
```

```
age              int64
workclass       object
education       object
educationno      int64
maritalstatus   object
occupation      object
relationship    object
race            object
sex             object
capitalgain      int64
capitalloss      int64
hoursperweek     int64
native          object
Salary          object
dtype: object
```

```
catg=sal_test.select_dtypes("object")
cont=sal_test.select_dtypes("int")
catg.head()
catg.shape
```

```
(15060, 9)
```

```
from sklearn.preprocessing import LabelEncoder
LE=LabelEncoder()
for i in range(0,9):
  catg.iloc[:,i]=LE.fit_transform(catg.iloc[:,i])
catg.head()
catg1=catg
catg1.head()
```

```
/usr/local/lib/python3.8/dist-packages/pandas/core/indexing.py:1773: SettingWithCo
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  self._setitem_single_column(ilocs[0], value, pi)
```

	workclass	education	maritalstatus	occupation	relationship	race	sex	nativ
1	2	11	2	4	0	4	1	3

####concating the both cont and catg variables

```
df=pd.concat([cont,catg1],axis=1)
df.head()
```

	age	educationno	capitalgain	capitalloss	hoursperweek	workclass	education
0	25	7	0	0	40	2	1
1	38	9	0	0	50	2	11
2	28	12	0	0	40	1	7
3	44	10	7688	0	40	2	15
4	34	6	0	0	30	2	0

```
df.shape
df1=df.drop(df.columns[[3,4]],axis=1)
df1.shape
df1.head()
```

	age	educationno	capitalgain	workclass	education	maritalstatus	occupation
0	25	7	0	2	1	4	6
1	38	9	0	2	11	2	4
2	28	12	0	1	7	2	10
3	44	10	7688	2	15	2	6
4	34	6	0	2	0	4	7

```
X=df.iloc[:,0:11]
Y=df["Salary"]
X.head()
```

	age	educationno	capitalgain	capitalloss	hoursperweek	workclass	education
0	25	7	0	0	40	2	1
1	38	9	0	0	50	2	11
2	28	12	0	0	40	1	7
3	44	10	7688	0	40	2	15
4	34	6	0	0	30	2	0

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y)
```

```
from sklearn.naive_bayes import MultinomialNB
MNB= MultinomialNB()
MNB.fit(X_test,Y_test)
```

```
MultinomialNB()
```

```
Y_pred_test=MNB.predict(X_test)
Y_pred_test
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

```
from sklearn.metrics import accuracy_score
test_accuracy=accuracy_score(Y_test,Y_pred_test).round(2)
test_accuracy
```

```
0.77
```

```
#############################################################################################
#############################################################################################
```

accessing train data **WORKING ON TRAIN DATA SET**

```
from google.colab import files
upload=files.upload()
```

Choose Files SalaryData_Train.csv

- **SalaryData_Train.csv**(text/csv) - 3393618 bytes, last modified: 2/24/2023 - 100% done

```
Saving SalaryData_Train.csv to SalaryData_Train.csv
```

```
import pandas as pd
sal_train=pd.read_csv("SalaryData_Train.csv")
sal_train.head()
```

	age	workclass	education	educationno	maritalstatus	occupation	relationship
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-family
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husband
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife

```
sal_train.isnull().sum()
sal_train.dtypes
```

```
age              int64
workclass       object
education       object
educationno      int64
maritalstatus   object
occupation      object
relationship    object
race            object
sex             object
capitalgain      int64
capitalloss      int64
hoursperweek     int64
native          object
Salary          object
dtype: object
```

```
catg=sal_train.select_dtypes("object")
cont=sal_train.select_dtypes("int")
catg.head()
catg.shape
```

```
(30161, 9)
```

```
from sklearn.preprocessing import LabelEncoder
LE=LabelEncoder()
for i in range(0,9):
  catg.iloc[:,i]=LE.fit_transform(catg.iloc[:,i])
catg.head()
catg.shape
```

```
/usr/local/lib/python3.8/dist-packages/pandas/core/indexing.py:1773: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  self._setitem_single_column(ilocs[0], value, pi)
(30161, 9)
```

```
df=pd.concat([cont,catg],axis=1)
df.head()
```

	age	educationno	capitalgain	capitalloss	hoursperweek	workclass	education
0	39	13	2174	0	40	5	9
1	50	13	0	0	13	4	9
2	38	9	0	0	40	2	11
3	53	7	0	0	40	2	1
4	28	13	0	0	40	2	9

```
df.shape
df1=df.drop(df.columns[[3,4]],axis=1)
df1.shape
df1.head()
```

	age	educationno	capitalgain	workclass	education	maritalstatus	occupation
0	39	13	2174	5	9	4	0
1	50	13	0	4	9	2	3
2	38	9	0	2	11	0	5
3	53	7	0	2	1	2	5
4	28	13	0	2	9	2	9

```
X=df.iloc[:,0:11]
Y=df["Salary"]
X.head()
X.shape
```

```
(30161, 11)
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y)
```

```
from sklearn.naive_bayes import MultinomialNB
MNB= MultinomialNB()
MNB.fit(X_train,Y_train)
```

```
MultinomialNB()
```

```
Y_pred_train=MNB.predict(X_train)
Y_pred_train
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

```
from sklearn.metrics import accuracy_score
train_accuracy=accuracy_score(Y_train,Y_pred_train).round(2)
train_accuracy
```

```
0.77
```

✓ 0s completed at 10:21 PM