Import Packages

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from scipy.stats import zscore

%matplotlib inline
```

```
# Import Data
data = pd.read_csv('adult.csv')
```

Quality Check

```
data.shape
```

```
(32560, 15)
```

```
data.head()
```

	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174
0	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0
1	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0

```
#Name of the columns
data.columns = ['Age', 'Workclass', 'fnlgt', 'Education', 'Education_num', 'Marital_Status', 'Occupation'
                ,'Relationship', 'Race', 'Sex', 'Gain', 'Loss', 'Hoursperweek', 'Country', 'Label']
```

```
data.head()
```

	Age	Workclass	fnlgt	Education	Education_num	Marital_Status	Occupation	Relatio
0	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Hu
1	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-
2	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Hu

```
# Data Types
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32560 entries, 0 to 32559
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Age             32560 non-null  int64 
 1   Workclass       32560 non-null  object
 2   fnlgt           32560 non-null  int64 
 3   Education       32560 non-null  object
 4   Education_num   32560 non-null  int64 
 5   Marital_Status  32560 non-null  object
```

```
 6   Occupation      32560 non-null  object
 7   Relationship    32560 non-null  object
 8   Race            32560 non-null  object
 9   Sex             32560 non-null  object
 10  Gain            32560 non-null  int64
 11  Loss            32560 non-null  int64
 12  Hoursperweek    32560 non-null  int64
 13  Country         32560 non-null  object
 14  Label           32560 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

Great! We don't have null values. Yes, my first thought! But.....

Change label to 0 and 1

le = LabelEncoder() data['Label'] = le.fit_transform(data['Label'])

```
le = LabelEncoder()
data['Label'] = le.fit_transform(data['Label'])
```

```
data.Label.value_counts()#/data.shape[0]
```

```
0    24719
1     7841
Name: Label, dtype: int64
```

Preprocessing

```
# Normalize the numerical data

num = [var for var in data.columns if (data[var].dtype == 'int64') & (var != 'Label')]
cat = [var for var in data.columns if data[var].dtype == 'object']
```

```
num
```

```
['Age', 'fnlgt', 'Education_num', 'Gain', 'Loss', 'Hoursperweek']
```

```
normalized_data = zscore(data[num], axis=1)
```

```
normalized_data.head()
```

	Age	fnlgt	Education_num	Gain	Loss	Hoursperweek
0	-0.446092	2.236068	-0.447284	-0.447703	-0.447703	-0.447284
1	-0.446957	2.236068	-0.447318	-0.447430	-0.447430	-0.446932
2	-0.446836	2.236068	-0.447362	-0.447442	-0.447442	-0.446985
3	-0.447120	2.236068	-0.447239	-0.447342	-0.447342	-0.447025
4	-0.447036	2.236068	-0.447253	-0.447385	-0.447385	-0.447008

```
## QC
np.mean(normalized_data), np.std(normalized_data)
```

```
(Age             -0.450492
 fnlgt            2.232898
 Education_num   -0.451105
 Gain            -0.431391
 Loss            -0.449455
 Hoursperweek    -0.450455
 dtype: float64,
 Age              0.019186
 fnlgt            0.046391
 Education_num    0.019260
 Gain             0.113947
 Loss             0.021548
```

```
Hoursperweek     0.019194
dtype: float64)
```

```
data_v1 = pd.concat([normalized_data, data[cat], data['Label']], axis = 1)
```

```
data_v1.head()
```

	Age	fnlgt	Education_num	Gain	Loss	Hoursperweek	Workclass	Education	Marital_Status	Occupation	Relationship
0	-0.446092	2.236068	-0.447284	-0.447703	-0.447703	-0.447284	Self-emp-not-inc	Bachelors	Married-civ-spouse	Exec-managerial	Husband
1	-0.446957	2.236068	-0.447318	-0.447430	-0.447430	-0.446932	Private	HS-grad	Divorced	Handlers-cleaners	Not-in-family
2	-0.446836	2.236068	-0.447362	-0.447442	-0.447442	-0.446985	Private	11th	Married-civ-spouse	Handlers-cleaners	Husband

```
data_v1.shape
```

```
(32560, 15)
```

```
# Categorical varibales
cat
```

```
['Workclass',
 'Education',
 'Marital_Status',
 'Occupation',
 'Relationship',
 'Race',
 'Sex',
 'Country']
```

```
data_v1.Workclass.value_counts()
```

```
 Private             22696
 Self-emp-not-inc     2541
 Local-gov            2093
 ?                    1836
 State-gov            1297
 Self-emp-inc         1116
 Federal-gov           960
 Without-pay            14
 Never-worked            7
Name: Workclass, dtype: int64
```

.....But, we do have Null values in our data, just not in a standard format. Let's deal with them! Let's check for other categorical columns as well and replace these values with NaN

```
data_v1['Workclass'].replace('?', np.NaN, inplace=True)
```

```
data_v1.Workclass.value_counts()
```

```
 Private             22696
 Self-emp-not-inc     2541
 Local-gov            2093
 ?                    1836
 State-gov            1297
 Self-emp-inc         1116
 Federal-gov           960
 Without-pay            14
 Never-worked            7
Name: Workclass, dtype: int64
```

Strange! It seems like they are not strings, so lets convert them to strings

```
data_v1['Workclass'] = data_v1['Workclass'].astype('string')
```

```
data_v1['Workclass'].replace('?', np.NaN, inplace=True)
```

```
data_v1.Workclass.value_counts()
```

```
Private             22696
Self-emp-not-inc     2541
Local-gov            2093
?                    1836
State-gov            1297
Self-emp-inc         1116
Federal-gov           960
Without-pay            14
Never-worked            7
Name: Workclass, dtype: Int64
```

Not working, Now there's only one thing left to do. remove these rows

```
rows = data_v1['Workclass'].str.contains(r"\?")
```

```
data_v1= data_v1.drop(rows[rows==True].index.to_list(), axis=0)
```

```
data_v1.shape
```

```
(30724, 15)
```

```
#for var in cat:
   # print(data_v1[var].value_counts())
```

```
rows = data_v1['Occupation'].str.contains(r"\?")
```

```
data_v1= data_v1.drop(rows[rows==True].index.to_list(), axis=0)
```

```
data_v1.shape
```

```
(30717, 15)
```

```
rows = data_v1['Country'].str.contains(r"\?")
```

```
data_v1= data_v1.drop(rows[rows==True].index.to_list(), axis=0)
```

```
data_v1.shape
```

```
(30161, 15)
```

```
data_v1.isna().sum()
```

```
Age               0
fnlgt             0
Education_num     0
Gain              0
Loss              0
Hoursperweek      0
Workclass         0
Education         0
Marital_Status    0
Occupation        0
Relationship      0
Race              0
Sex               0
Country           0
Label             0
dtype: int64
```

Check for Cardinality

```
for var in cat:
    print(var+' contains '+str(data_v1[var].nunique())+' labels')
    print()
```

```
Workclass contains 7 labels

Education contains 16 labels

Marital_Status contains 7 labels

Occupation contains 14 labels
```

```
Relationship contains 6 labels

Race contains 5 labels

Sex contains 2 labels

Country contains 41 labels
```

```
data_v1['Country'].value_counts()
```

```
 United-States                 27503
 Mexico                          610
 Philippines                     188
 Germany                         128
 Puerto-Rico                     109
 Canada                          107
 India                           100
 El-Salvador                     100
 Cuba                             92
 England                          86
 Jamaica                          80
 South                            71
 China                            68
 Italy                            68
 Dominican-Republic               67
 Vietnam                          64
 Guatemala                        63
 Japan                            59
 Poland                           56
 Columbia                         56
 Iran                             42
 Taiwan                           42
 Haiti                            42
 Portugal                         34
 Nicaragua                        33
 Peru                             30
 Greece                           29
 France                           27
 Ecuador                          27
 Ireland                          24
 Hong                             19
 Cambodia                         18
 Trinadad&Tobago                  18
 Thailand                         17
 Laos                             17
 Yugoslavia                       16
 Outlying-US(Guam-USVI-etc)       14
 Hungary                          13
 Honduras                         12
 Scotland                         11
 Holand-Netherlands                1
Name: Country, dtype: int64
```

91% of the values are US, so we'll club all other countries in Other

```
list_of_countries = [Country for Country in data_v1.Country.unique() if Country != ' United-States']
```

```
data_v1['Country'] = data_v1['Country'].apply(lambda x: 'Other' if x in list_of_countries else x)
```

```
data_v1.Country.value_counts()
```

```
 United-States    27503
Other              2658
Name: Country, dtype: int64
```

```
cat
```

```
['Workclass',
 'Education',
 'Marital_Status',
 'Occupation',
 'Relationship',
 'Race',
 'Sex',
 'Country']
```

```
# One Hot Encoding

data_final = pd.get_dummies(data_v1, columns = cat)

data_final.head()
```

	Age	fnlgt	Education_num	Gain	Loss	Hoursperweek	Label	Workclass_ Federal-gov	Workclass_ Local-gov	Workclass_ Private	...	Relationship_ Wife	In E
0	-0.446092	2.236068	-0.447284	-0.447703	-0.447703	-0.447284	0	0	0	0	...	0	
1	-0.446957	2.236068	-0.447318	-0.447430	-0.447430	-0.446932	0	0	0	1	...	0	
2	-0.446836	2.236068	-0.447362	-0.447442	-0.447442	-0.446985	0	0	0	1	...	0	
3	-0.447120	2.236068	-0.447239	-0.447342	-0.447342	-0.447025	0	0	0	1	...	1	
4	-0.447036	2.236068	-0.447253	-0.447385	-0.447385	-0.447008	0	0	0	1	...	1	

5 rows × 66 columns

```
data_final.shape
```

```
(30161, 66)
```

```
data_final.info()
```

```
 10  Workclass_ Self-emp-inc              30161 non-null  uint8
 11  Workclass_ Self-emp-not-inc          30161 non-null  uint8
 12  Workclass_ State-gov                 30161 non-null  uint8
 13  Workclass_ Without-pay               30161 non-null  uint8
 14  Education_ 10th                      30161 non-null  uint8
 15  Education_ 11th                      30161 non-null  uint8
 16  Education_ 12th                      30161 non-null  uint8
 17  Education_ 1st-4th                   30161 non-null  uint8
 18  Education_ 5th-6th                   30161 non-null  uint8
 19  Education_ 7th-8th                   30161 non-null  uint8
 20  Education_ 9th                       30161 non-null  uint8
 21  Education_ Assoc-acdm                30161 non-null  uint8
 22  Education_ Assoc-voc                 30161 non-null  uint8
 23  Education_ Bachelors                 30161 non-null  uint8
 24  Education_ Doctorate                 30161 non-null  uint8
 25  Education_ HS-grad                   30161 non-null  uint8
 26  Education_ Masters                   30161 non-null  uint8
 27  Education_ Preschool                 30161 non-null  uint8
 28  Education_ Prof-school               30161 non-null  uint8
 29  Education_ Some-college              30161 non-null  uint8
 30  Marital_Status_ Divorced             30161 non-null  uint8
 31  Marital_Status_ Married-AF-spouse    30161 non-null  uint8
 32  Marital_Status_ Married-civ-spouse   30161 non-null  uint8
 33  Marital_Status_ Married-spouse-absent  30161 non-null  uint8
 34  Marital_Status_ Never-married        30161 non-null  uint8
 35  Marital_Status_ Separated            30161 non-null  uint8
 36  Marital_Status_ Widowed              30161 non-null  uint8
 37  Occupation_ Adm-clerical             30161 non-null  uint8
 38  Occupation_ Armed-Forces             30161 non-null  uint8
 39  Occupation_ Craft-repair             30161 non-null  uint8
 40  Occupation_ Exec-managerial          30161 non-null  uint8
 41  Occupation_ Farming-fishing          30161 non-null  uint8
 42  Occupation_ Handlers-cleaners        30161 non-null  uint8
 43  Occupation_ Machine-op-inspct        30161 non-null  uint8
 44  Occupation_ Other-service            30161 non-null  uint8
 45  Occupation_ Priv-house-serv          30161 non-null  uint8
```

```
 61  Race_ White                      30161 non-null  uint8
 62  Sex_ Female                      30161 non-null  uint8
 63  Sex_ Male                        30161 non-null  uint8
 64  Country_ United-States           30161 non-null  uint8
 65  Country_Other                    30161 non-null  uint8
dtypes: float64(6), int64(1), uint8(59)
memory usage: 3.5 MB
```

Splitting the Data

```
X = data_final.drop(columns = 'Label')
Y = data_final['Label']
```

```
X.shape, Y.shape
```

```
((30161, 65), (30161,))
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.1, random_state = 42)
```

```
X_train.shape, X_test.shape
```

```
((27144, 65), (3017, 65))
```

```
X_train.head()
```

	Age	fnlgt	Education_num	Gain	Loss	Hoursperweek	Workclass_ Federal-gov	Workclass_ Local-gov	Workclass_ Private	Workclass_ Self-emp-inc	...	Relatior
31438	-0.447169	2.236068	-0.447241	-0.447291	-0.447291	-0.447076	0	0	1	0	...	
7280	-0.444323	2.236064	-0.448223	-0.449021	-0.449021	-0.445475	0	0	1	0	...	
31021	-0.446848	2.236068	-0.447341	-0.447442	-0.447442	-0.446994	0	0	1	0	...	
30953	-0.484874	2.230218	-0.486059	-0.287675	-0.486334	-0.485276	0	0	1	0	...	
24047	-0.446908	2.236068	-0.447284	-0.447444	-0.447444	-0.446988	0	0	0	0	...	

5 rows × 65 columns

Model Training

```
from sklearn.naive_bayes import GaussianNB

# Instantiate the model
gnb = GaussianNB()
```

```
# Fit to train data
gnb.fit(X_train, Y_train)
```

```
▾ GaussianNB
GaussianNB()
```

```
from sklearn.metrics import accuracy_score

acc_train = accuracy_score(Y_train, gnb.predict(X_train))
acc_test = accuracy_score(Y_test, gnb.predict(X_test))
```

```
acc_train, acc_test
```

```
(0.6971706454465075, 0.6920782234007292)
```

The training-set accuracy score is 0.697 while the test-set accuracy to be 0.711. These two values are quite comparable. So, there is no sign of overfitting.