```
In [1]: import pandas as pd
        import seaborn as sns
        import numpy as np
        import matplotlib.pyplot as plt

        %matplotlib inline

        #ignore warning
        import warnings
        warnings.filterwarnings('ignore')
```

```
In [2]: df = pd.read_csv(r"D:\Data Science with AI\2nd,3rd-jan-2024\Project\adult.csv\
```

```
In [3]: df
```

Out[3]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationshi
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-famil
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-famil
2	66	?	186061	Some-college	10	Widowed	?	Unmarrie
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarrie
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-chil
...	...	...	...	...	...	...	...	..
32556	22	Private	310152	Some-college	10	Never-married	Protective-serv	Not-in-famil
32557	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wif
32558	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husban
32559	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarrie
32560	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-chil

32561 rows × 15 columns

check dataset and shape

In [4]:
```
df.shape
```
Out[4]:
```
(32561, 15)
```

In [5]:
```
df.head()
```
Out[5]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	r
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-family	W
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	W
2	66	?	186061	Some-college	10	Widowed	?	Unmarried	B
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	W
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	W

In [6]:
```
df.columns
```
Out[6]:
```
Index(['age', 'workclass', 'fnlwgt', 'education', 'education.num',
       'marital.status', 'occupation', 'relationship', 'race', 'sex',
       'capital.gain', 'capital.loss', 'hours.per.week', 'native.country',
       'income'],
      dtype='object')
```

view summary dataframe

In [7]:
```
df.info()
```
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  object
 4   education.num   32561 non-null  int64
 5   marital.status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital.gain    32561 non-null  int64
 11  capital.loss    32561 non-null  int64
 12  hours.per.week  32561 non-null  int64
 13  native.country  32561 non-null  object
```

Encode ? as NANS

In [8]:
```
df[df =='?'] = np.nan
```

again check summary dataframe

In [9]:
```
df.shape
```

Out[9]:
```
(32561, 15)
```

impute missing values with mode

In [10]:
```
for col in ['workclass' ,'occupation','native.country']:
    df[col].fillna(df[col].mode()[0],inplace=True)
```

check again missing values

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
age               0
workclass         0
fnlwgt            0
education         0
education.num     0
marital.status    0
occupation        0
relationship      0
race              0
sex               0
capital.gain      0
capital.loss      0
hours.per.week    0
native.country    0
income            0
dtype: int64
```

setting feature vector and target variable

In [12]:
```
x=df.drop(['income'],axis=1)
y=df['income']
```

In [13]:
```
x.head
```

Out[13]:
```
<bound method NDFrame.head of        age workclass  fnlwgt     education  edu
cation.num      marital.status  \
0       90   Private   77053       HS-grad              9             Widowed
1       82   Private  132870       HS-grad              9             Widowed
2       66   Private  186061  Some-college             10             Widowed
3       54   Private  140359       7th-8th              4            Divorced
4       41   Private  264663  Some-college             10           Separated
...    ...       ...     ...           ...            ...                 ...
32556   22   Private  310152  Some-college             10       Never-married
32557   27   Private  257302    Assoc-acdm             12  Married-civ-spouse
32558   40   Private  154374       HS-grad              9  Married-civ-spouse
32559   58   Private  151910       HS-grad              9             Widowed
32560   22   Private  201490       HS-grad              9       Never-married

              occupation   relationship   race     sex  capital.gain  \
0         Prof-specialty  Not-in-family  White  Female             0
1        Exec-managerial  Not-in-family  White  Female             0
2         Prof-specialty      Unmarried  Black  Female             0
3      Machine-op-inspct      Unmarried  White  Female             0
4         Prof-specialty      Own-child  White  Female             0
...                  ...            ...    ...     ...           ...
32556    Protective-serv  Not-in-family  White    Male             0
32557       Tech-support           Wife  White  Female             0
32558  Machine-op-inspct        Husband  White    Male             0
32559       Adm-clerical      Unmarried  White  Female             0
32560       Adm-clerical      Own-child  White    Male             0

       capital.loss  hours.per.week native.country
0              4356              40  United-States
1              4356              18  United-States
2              4356              40  United-States
3              3900              40  United-States
4              3900              40  United-States
...             ...             ...            ...
32556             0              40  United-States
32557             0              38  United-States
32558             0              40  United-States
32559             0              40  United-States
32560             0              20  United-States

[32561 rows x 14 columns]>
```

split the data into traing and testing set

In [14]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.20,random_stat
```

feature engineering

#encode categorical categorical variable

In [15]:
```
from sklearn import preprocessing
categorical=['workclass', 'education', 'marital.status', 'occupation', 'relati
for feature in categorical :
    le=preprocessing.LabelEncoder()
    x_train[feature]=le.fit_transform(x_train[feature])
    x_test[feature]=le.transform(x_test[feature])
```

feature scaling

In [16]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=pd.DataFrame(sc.fit_transform(x_train),columns=x.columns)
x_test=pd.DataFrame(sc.transform(x_test),columns=x.columns)
```

In [17]:
```
x_train.head(10)
```

Out[17]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relation
0	1.039817	-2.817895	0.960050	1.212397	-0.041721	-1.745708	-1.534809	-0.27
1	-1.407472	-0.085865	-0.461075	1.212397	-0.041721	0.915041	0.746328	0.96
2	-0.327786	-0.085865	-0.192395	0.177344	-0.429845	-1.745708	-0.014051	1.58
3	-1.047576	-0.085865	2.042380	0.177344	-0.429845	-0.415333	-1.027890	-0.90
4	-0.471744	-0.085865	-0.785647	0.177344	-0.429845	0.915041	0.239409	-0.27
5	0.895859	1.735487	-0.613820	0.436107	1.510776	-0.415333	-0.774430	-0.90
6	-0.111848	-0.085865	-0.027784	0.177344	-0.429845	0.915041	1.253248	1.58
7	0.248047	-0.085865	0.439758	-0.857710	0.734528	-0.415333	-1.027890	-0.90
8	1.327733	-0.085865	-1.143935	0.177344	-0.429845	-0.415333	1.253248	-0.90
9	-1.407472	-0.085865	1.526412	-2.669054	-1.594218	0.915041	-0.014051	0.96

logestic regression model with all features

In [18]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
lg=LogisticRegression()
lg.fit(x_train,y_train)
y_pred=lg.predict(x_test)
print('Logistic Regression accuracy score with all the features: {0:0.4f}'.for
```

```
Logistic Regression accuracy score with all the features: 0.8220
```

Explian variance ratio

In [19]:

```
from sklearn.decomposition import PCA
pca = PCA()
x_train = pca.fit_transform(x_train)
pca.explained_variance_ratio_
```

Out[19]:

```
array([0.14913354, 0.10195351, 0.08211085, 0.0799947 , 0.07478586,
       0.07344007, 0.06851712, 0.0666489 , 0.06371768, 0.06181611,
       0.05989432, 0.04841688, 0.04200177, 0.02756869])
```

logestic regression with first 13 feature

In [20]:

```
x=df.drop(['income','native.country'] , axis=1)
y=df['income']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=
categorical = ['workclass', 'education', 'marital.status', 'occupation', 'rela
for feature in categorical:
    le=preprocessing.LabelEncoder()
    x_train[feature]=le.fit_transform(x_train[feature])
    x_test[feature]=le.transform(x_test[feature])

x_train=pd.DataFrame(sc.fit_transform(x_train),columns=x.columns)
x_test=pd.DataFrame(sc.transform(x_test),columns=x.columns)
logreg=LogisticRegression()
logreg.fit(x_train,y_train)
y_pred=logreg.predict(x_test)
```

In [21]:

```
#logestic regression 12 feature
x=df.drop(['income','native.country', 'hours.per.week'] ,axis=1)
y=df['income']


x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_st
categorical=['workclass', 'education', 'marital.status', 'occupation', 're
for feature in categorical:
    le=preprocessing.LabelEncoder()
    x_train[feature]=le.fit_transform(x_train[feature])
    x_test[feature]=le.transform(x_test[feature])
x_train=pd.DataFrame(sc.fit_transform(x_train),columns=x.columns)

x_test=pd.DataFrame(sc.transform(x_test),columns=x.columns)

logreg=LogisticRegression()
logreg.fit(x_train,y_train)
y_pred=logreg.predict(x_test)
```

In [24]:

```
# logestic regression with first 11 feature

x=df.drop(['income','native.country', 'hours.per.week', 'capital.loss'] ,
y=df['income']

x_train,x_test,y_train,y_test=train_test_split(x,y, test_size=0.3, random_

categorical=['workclass', 'education', 'marital.status', 'occupation', 're
for feature in categorical:
    le=preprocessing.LabelEncoder()
    x_train[feature]=le.fit_transform(x_train[feature])
    x_test[feature]=le.transform(x_test[feature])

x_train=pd.DataFrame(sc.fit_transform(x_train), columns=x.columns)
x_test=pd.DataFrame(sc.transform(x_test), columns=x.columns)

logreg=LogisticRegression()
logreg.fit(x_train,y_train)
y_pred=logreg.predict(x_test)
```

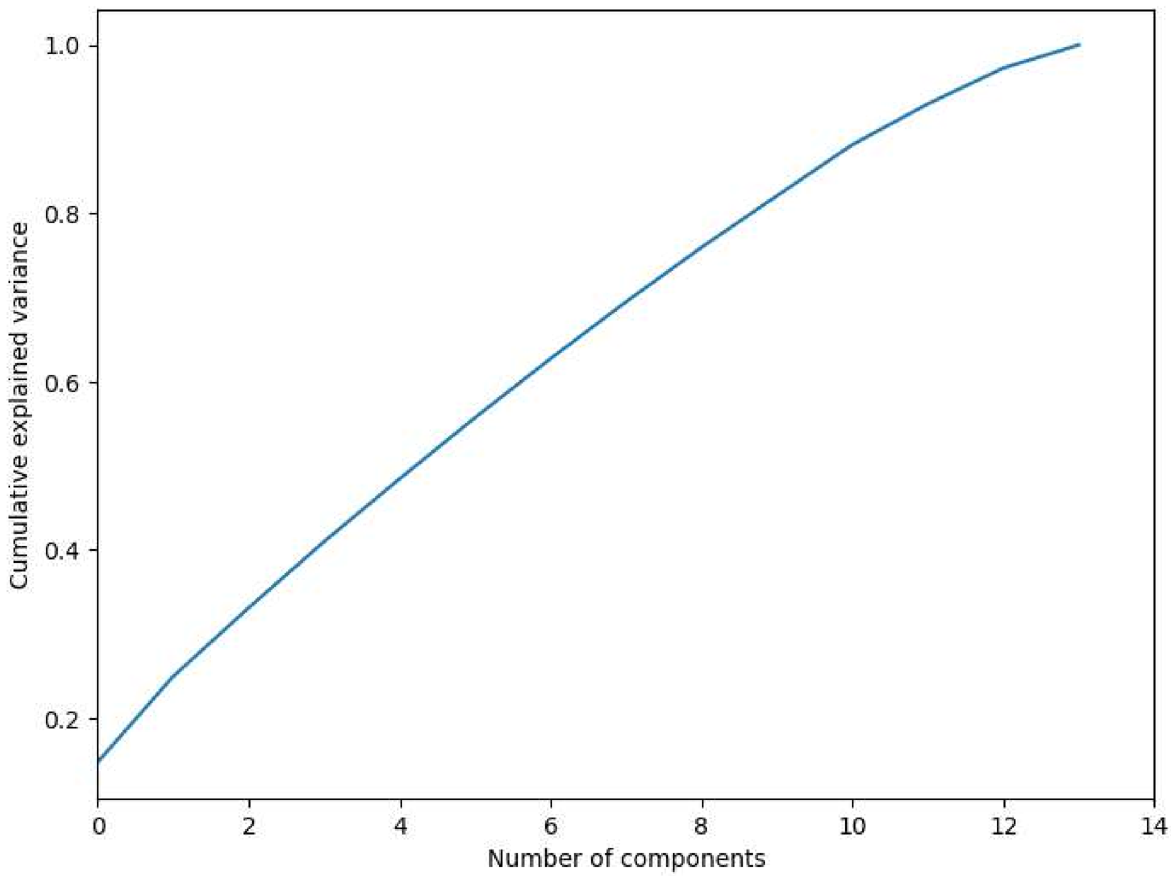
In [32]:

```
#select number of dimension

x=df.drop(['income'] ,axis=1)
y=df['income']

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=

categorical=['workclass', 'education', 'marital.status', 'occupation', 'relati
for feature in categorical:

    le = preprocessing.LabelEncoder()
    x_train[feature] = le.fit_transform(x_train[feature])
    x_test[feature] = le.transform(x_test[feature])


x_train = pd.DataFrame(sc.fit_transform(x_train), columns = x.columns)


pca= PCA()
pca.fit(x_train)
cumsum = np.cumsum(pca.explained_variance_ratio_)
dim = np.argmax(cumsum >= 0.90) + 1
print('The number of dimensions required to preserve 90% of variance is',dim)
```

```
The number of dimensions required to preserve 90% of variance is 12
```

In [34]:

```python
#plot explained variance ratio with number of dimensions
plt.figure(figsize=(8,6))
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlim(0,14,1)
plt.xlabel('Number of components')
plt.ylabel('Cumulative explained variance')
plt.show()
```



In [ ]: