UNIT 3.1 GRADED ASSIGNMENT

Group members

Ifra Saleem (2303.khi.deg.003)
Umaima Siddiqui (2023.KHI.DEG.033)

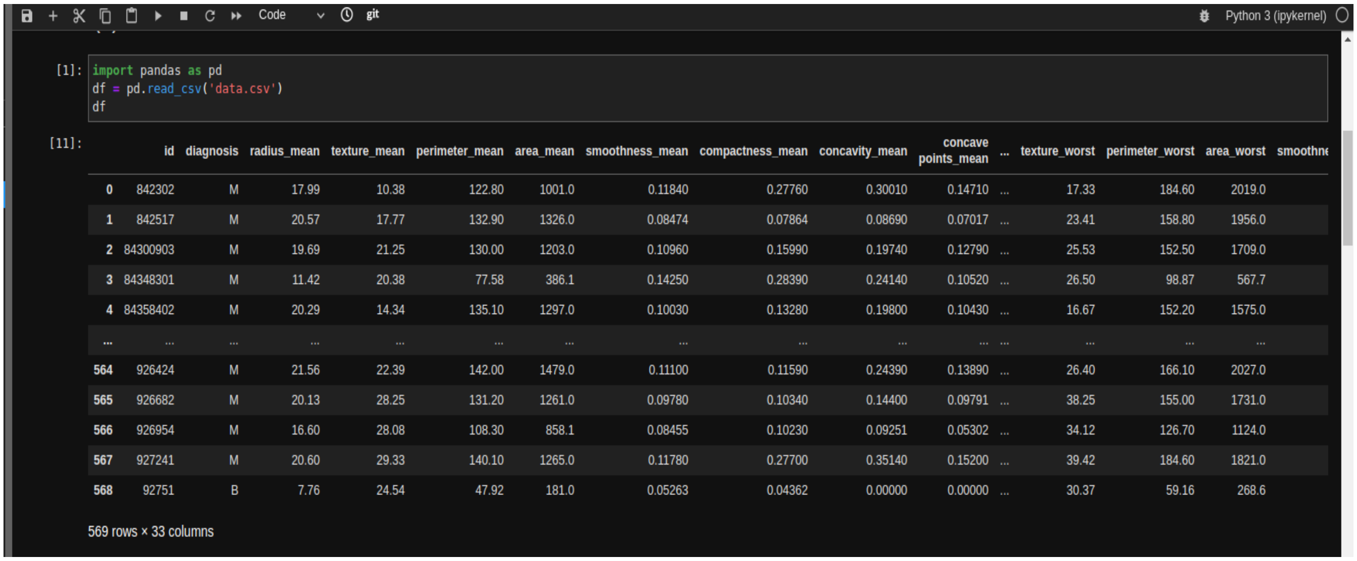
UNIT 3.1 GRADED ASSIGNMENT

Task:

Implement a label encoder for categorical data using pure Python, Pandas and NumPy.

Solution:

1) Label Encoding on breast cancer dataset, on a single column (diagnosis):

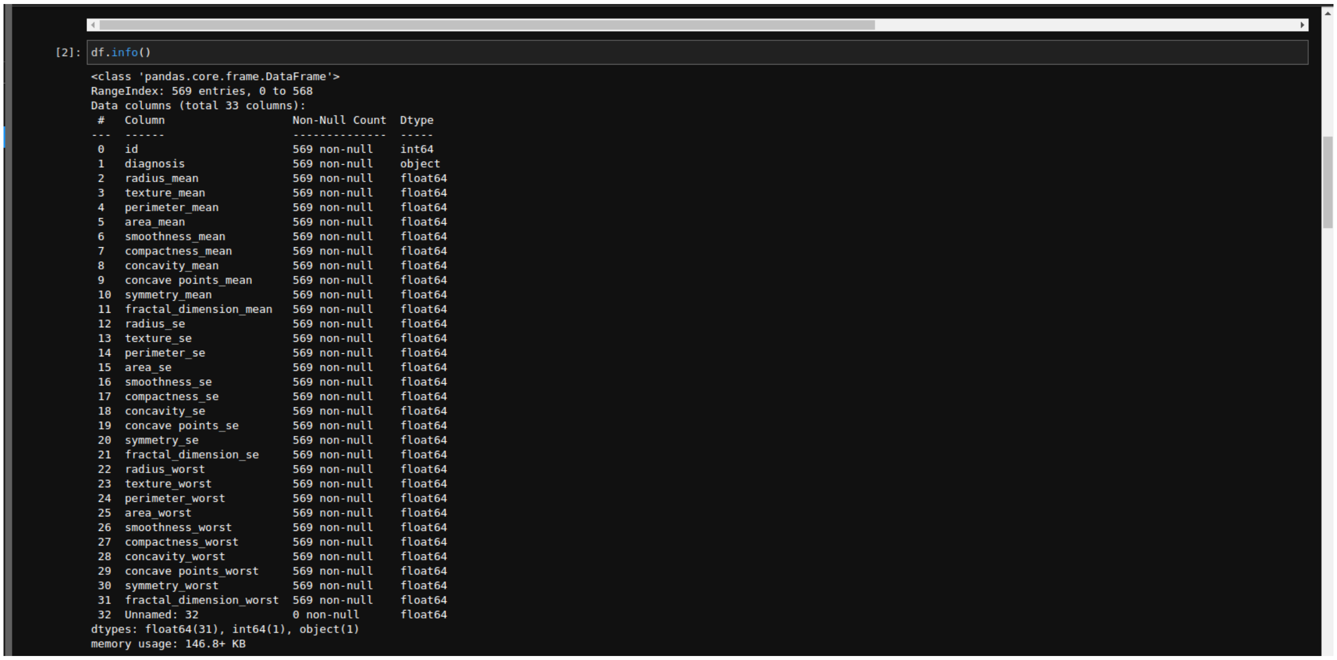


```
[1]: import pandas as pd
     df = pd.read_csv('data.csv')
     df
```

[11]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	texture_worst	perimeter_worst	area_worst	smoothne
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	...	17.33	184.60	2019.0	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	...	23.41	158.80	1956.0	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	...	25.53	152.50	1709.0	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	...	26.50	98.87	567.7	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	...	16.67	152.20	1575.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	...	26.40	166.10	2027.0	
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...	38.25	155.00	1731.0	
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	...	34.12	126.70	1124.0	
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	...	39.42	184.60	1821.0	
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	...	30.37	59.16	268.6	

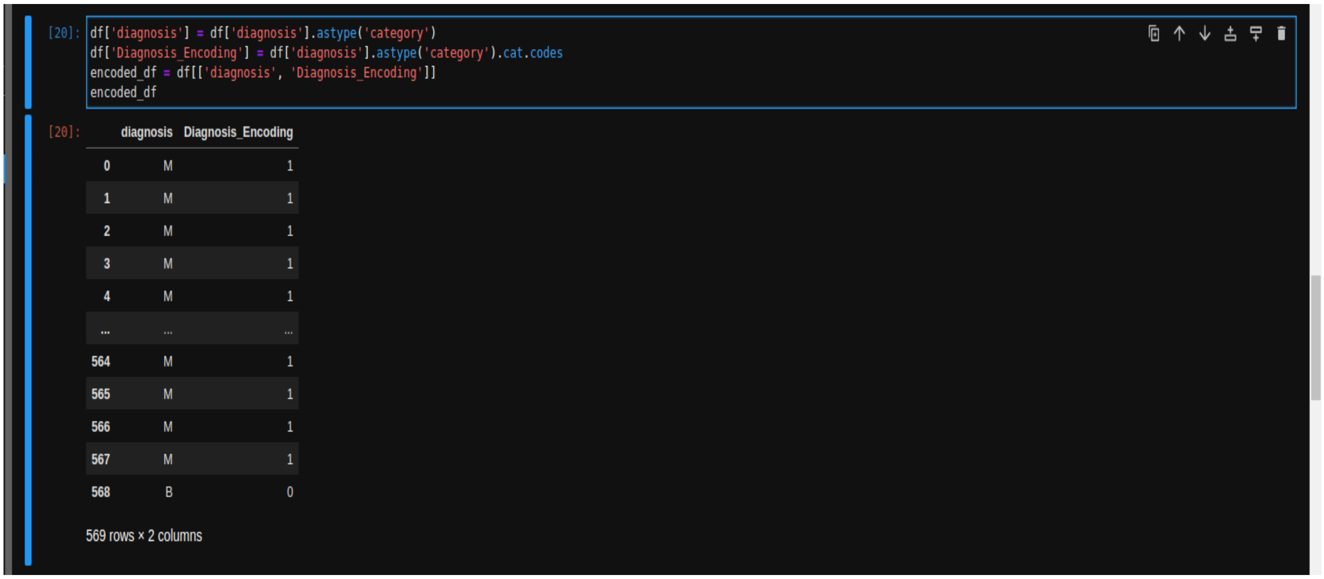
569 rows × 33 columns



```
[2]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    object 
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

Output:



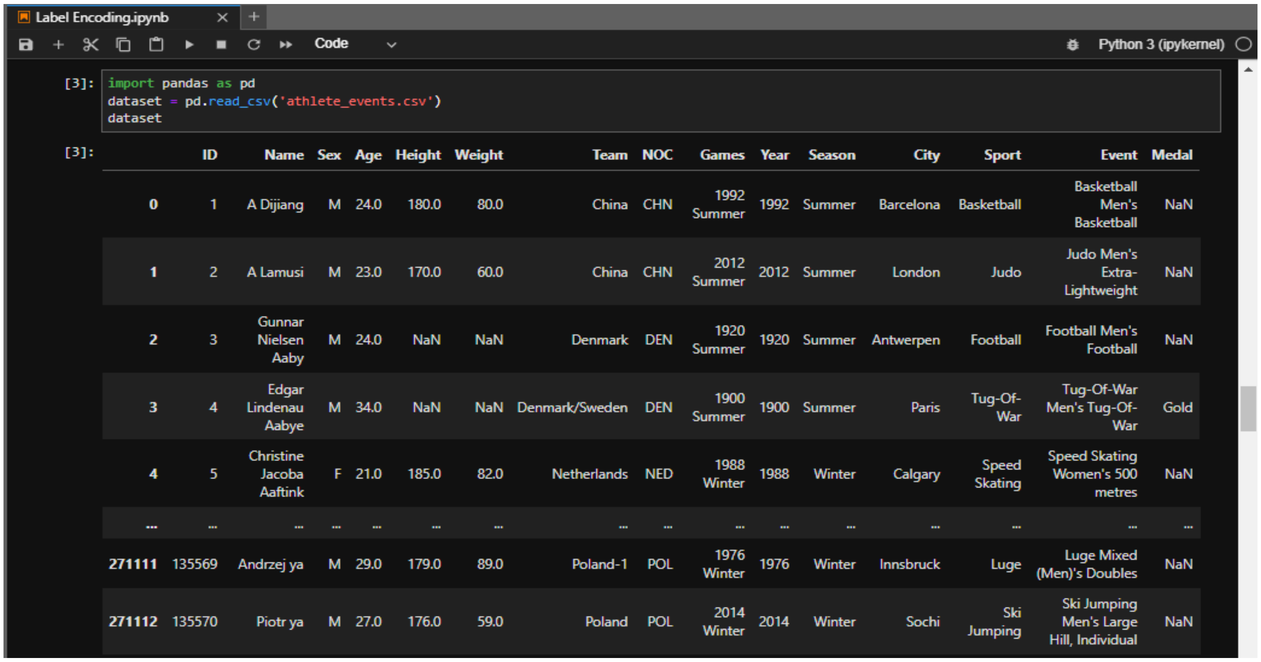
```
[20]: df['diagnosis'] = df['diagnosis'].astype('category')
      df['Diagnosis_Encoding'] = df['diagnosis'].astype('category').cat.codes
      encoded_df = df[['diagnosis', 'Diagnosis_Encoding']]
      encoded_df
```

[20]:

	diagnosis	Diagnosis_Encoding
0	M	1
1	M	1
2	M	1
3	M	1
4	M	1
...	...	...
564	M	1
565	M	1
566	M	1
567	M	1
568	B	0

569 rows × 2 columns

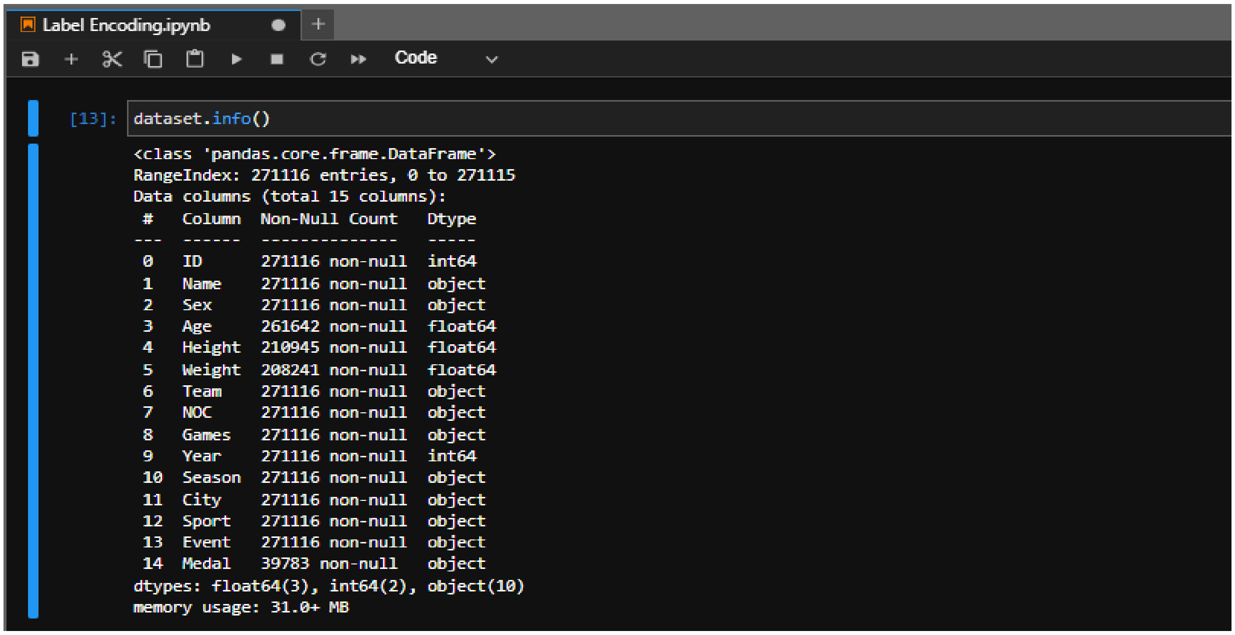
2) Label Encoding on athlete_events dataset, on multiple columns using a label_encoder function:



```
[3]: import pandas as pd
     dataset = pd.read_csv('athlete_events.csv')
     dataset
```

[3]:

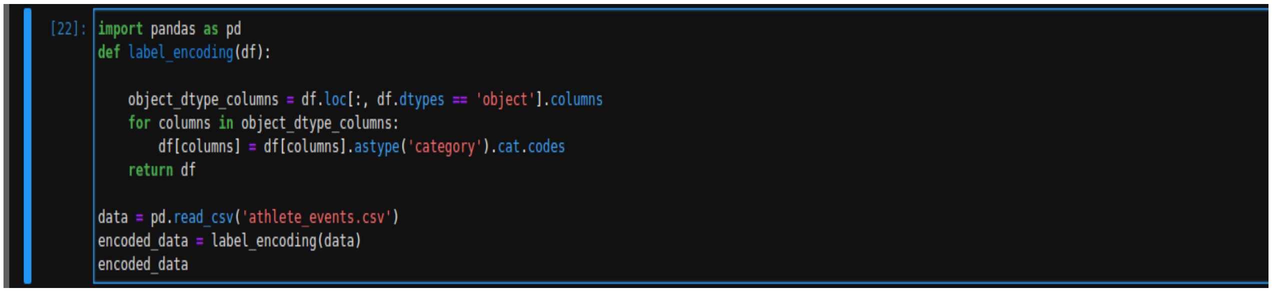
	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
271111	135569	Andrzej ya	M	29.0	179.0	89.0	Poland-1	POL	1976 Winter	1976	Winter	Innsbruck	Luge	Luge Mixed (Men)'s Doubles	NaN
271112	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winter	Sochi	Ski Jumping	Ski Jumping Men's Large Hill, Individual	NaN



```
[13]: dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271116 entries, 0 to 271115
Data columns (total 15 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   ID      271116 non-null  int64  
 1   Name    271116 non-null  object 
 2   Sex     271116 non-null  object 
 3   Age     261642 non-null  float64
 4   Height  210945 non-null  float64
 5   Weight  208241 non-null  float64
 6   Team    271116 non-null  object 
 7   NOC     271116 non-null  object 
 8   Games   271116 non-null  object 
 9   Year    271116 non-null  int64  
 10  Season  271116 non-null  object 
 11  City    271116 non-null  object 
 12  Sport   271116 non-null  object 
 13  Event   271116 non-null  object 
 14  Medal   39783 non-null   object 
dtypes: float64(3), int64(2), object(10)
memory usage: 31.0+ MB
```

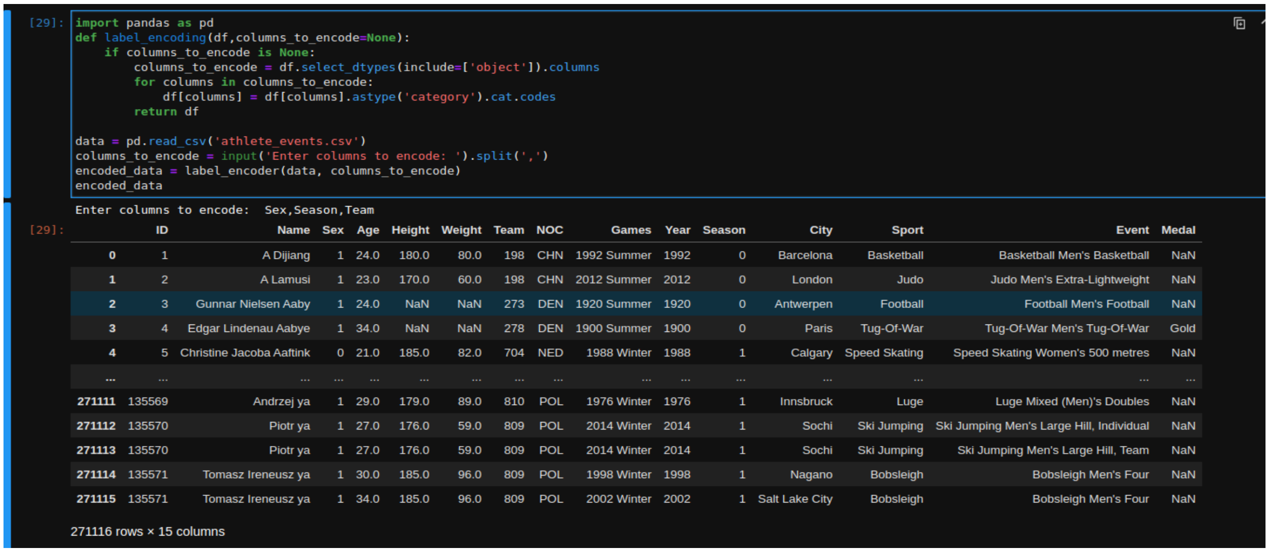
Label_encoder function:

This function takes a dataframe as parameter and it will first locate all the columns with the object data type in that dataframe and then it will convert each column with object data type into categorical data type using astype() method. After that it will perform label encoding on the columns with categorical data type.



```python
import pandas as pd
def label_encoding(df):

    object_dtype_columns = df.loc[:, df.dtypes == 'object'].columns
    for columns in object_dtype_columns:
        df[columns] = df[columns].astype('category').cat.codes
    return df

data = pd.read_csv('athlete_events.csv')
encoded_data = label_encoding(data)
encoded_data
```

If user gives input that which columns user wants to encode:



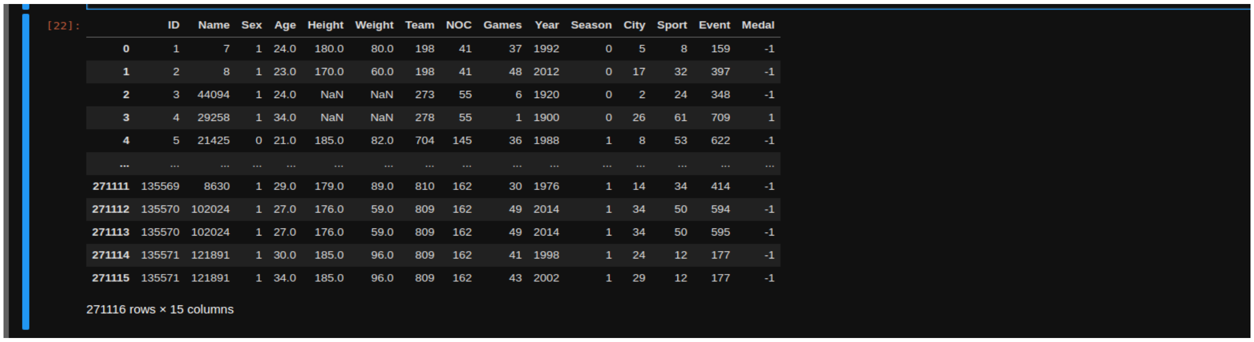
```python
import pandas as pd
def label_encoding(df,columns_to_encode=None):
    if columns_to_encode is None:
        columns_to_encode = df.select_dtypes(include=['object']).columns
        for columns in columns_to_encode:
            df[columns] = df[columns].astype('category').cat.codes
        return df

data = pd.read_csv('athlete_events.csv')
columns_to_encode = input('Enter columns to encode: ').split(',')
encoded_data = label_encoder(data, columns_to_encode)
encoded_data
```

```
Enter columns to encode:  Sex,Season,Team
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	1	24.0	180.0	80.0	198	CHN	1992 Summer	1992	0	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	1	23.0	170.0	60.0	198	CHN	2012 Summer	2012	0	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	1	24.0	NaN	NaN	273	DEN	1920 Summer	1920	0	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	1	34.0	NaN	NaN	278	DEN	1900 Summer	1900	0	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	0	21.0	185.0	82.0	704	NED	1988 Winter	1988	1	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
271111	135569	Andrzej ya	1	29.0	179.0	89.0	810	POL	1976 Winter	1976	1	Innsbruck	Luge	Luge Mixed (Men)'s Doubles	NaN
271112	135570	Piotr ya	1	27.0	176.0	59.0	809	POL	2014 Winter	2014	1	Sochi	Ski Jumping	Ski Jumping Men's Large Hill, Individual	NaN
271113	135570	Piotr ya	1	27.0	176.0	59.0	809	POL	2014 Winter	2014	1	Sochi	Ski Jumping	Ski Jumping Men's Large Hill, Team	NaN
271114	135571	Tomasz Ireneusz ya	1	30.0	185.0	96.0	809	POL	1998 Winter	1998	1	Nagano	Bobsleigh	Bobsleigh Men's Four	NaN
271115	135571	Tomasz Ireneusz ya	1	34.0	185.0	96.0	809	POL	2002 Winter	2002	1	Salt Lake City	Bobsleigh	Bobsleigh Men's Four	NaN

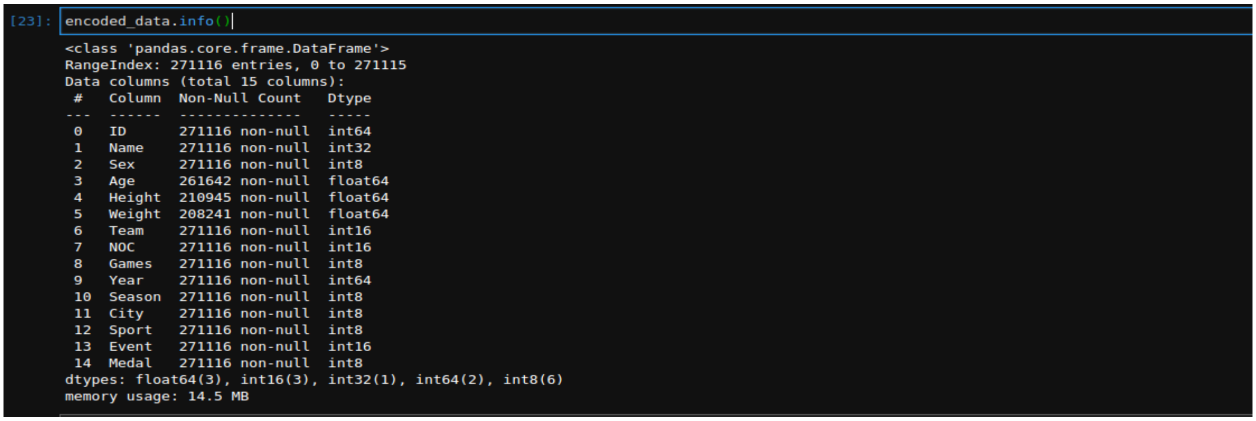
271116 rows × 15 columns

Output:



	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	7	1	24.0	180.0	80.0	198	41	37	1992	0	5	8	159	-1
1	2	8	1	23.0	170.0	60.0	198	41	48	2012	0	17	32	397	-1
2	3	44094	1	24.0	NaN	NaN	273	55	6	1920	0	2	24	348	-1
3	4	29258	1	34.0	NaN	NaN	278	55	1	1900	0	26	61	709	1
4	5	21425	0	21.0	185.0	82.0	704	145	36	1988	1	8	53	622	-1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
271111	135569	8630	1	29.0	179.0	89.0	810	162	30	1976	1	14	34	414	-1
271112	135570	102024	1	27.0	176.0	59.0	809	162	49	2014	1	34	50	594	-1
271113	135570	102024	1	27.0	176.0	59.0	809	162	49	2014	1	34	50	595	-1
271114	135571	121891	1	30.0	185.0	96.0	809	162	41	1998	1	24	12	177	-1
271115	135571	121891	1	34.0	185.0	96.0	809	162	43	2002	1	29	12	177	-1

271116 rows × 15 columns



```
[23]: encoded_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271116 entries, 0 to 271115
Data columns (total 15 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   ID      271116 non-null  int64  
 1   Name    271116 non-null  int32  
 2   Sex     271116 non-null  int8   
 3   Age     261642 non-null  float64
 4   Height  210945 non-null  float64
 5   Weight  208241 non-null  float64
 6   Team    271116 non-null  int16  
 7   NOC     271116 non-null  int16  
 8   Games   271116 non-null  int8   
 9   Year    271116 non-null  int64  
 10  Season  271116 non-null  int8   
 11  City    271116 non-null  int8   
 12  Sport   271116 non-null  int8   
 13  Event   271116 non-null  int16  
 14  Medal   271116 non-null  int8   
dtypes: float64(3), int16(3), int32(1), int64(2), int8(6)
memory usage: 14.5 MB
```