```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv("//content//data (2).csv")
df
```

	Unnamed: 0	id	age	salary	bought TV
0	0	1	51	8241.0	1
1	1	2	33	68076.0	0
2	2	3	41	7876.0	0
3	3	4	60	94066.0	1
4	4	5	58	37654.0	0
...	...	...	...	...	...
395	395	396	50	61700.0	1
396	396	397	45	2630.0	1
397	397	398	53	68894.0	1
398	398	399	NaN	16818.0	0
399	399	400	47	8294.0	1

400 rows × 5 columns

```
df.describe()
```

	Unnamed: 0	id	salary	bought TV
count	400.000000	400.000000	396.000000	400.000000
mean	199.500000	200.500000	27883.982323	0.275000
std	115.614301	115.614301	31236.075525	0.447073
min	0.000000	1.000000	-99999.000000	0.000000
25%	99.750000	100.750000	5563.750000	0.000000
50%	199.500000	200.500000	9351.000000	0.000000
75%	299.250000	300.250000	53482.000000	1.000000
max	399.000000	400.000000	98837.000000	1.000000

```
col_names = ['0', 'id', 'age', 'salary', 'boughttv',]
col_names
```

```
['0', 'id', 'age', 'salary', 'boughttv']
```

```
df.columns
```

```
Index(['Unnamed: 0', 'id', 'age', 'salary', 'bought TV'], dtype='object')
```

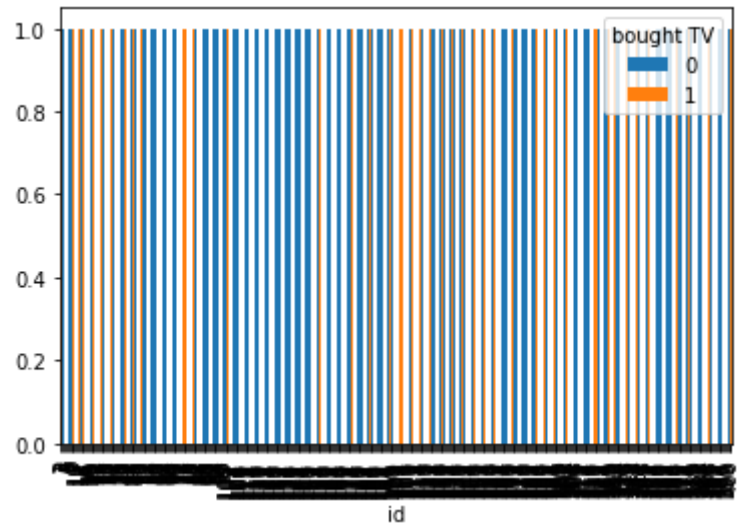
```
col_names = ['0', 'id', 'age', 'salary', 'boughttv',]
for col in col_names:

    print(df[col].value_counts())
```

```
df.isnull().sum()
```

```
Unnamed: 0    0
id            0
age           4
salary        4
bought TV     0
dtype: int64
```

```
df.groupby(['id','bought TV']).size().unstack().plot(kind='bar',stacked=True)
plt.show()
```




```
X = df.drop(['salary'], axis=1)
```

```
y = df['salary']
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state =
```

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

```
((268, 4), (132, 4))
```

```
X_train.dtypes
```

```
Unnamed: 0     int64
id             int64
age           object
bought TV      int64
dtype: object
```

```
from sklearn.tree import DecisionTreeClassifier
clf_en = DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=0)
clf_en.fit(X_train, y_train)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-36-14c42b663111> in <module>()
      2 from sklearn.tree import DecisionTreeClassifier
      3 clf_en = DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=0
----> 4 clf_en.fit(X_train, y_train)

 3 frames
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py in _assert_all_fi
msg_dtype)
     58                     msg_err.format
     59                     (type_err,
---> 60                      msg_dtype if msg_dtype is not None else X.dtype)
     61             )
     62     # for object dtype data, we only check for NaNs (GH-13254)

ValueError: Input contains NaN, infinity or a value too large for dtype('float32').
```

SEARCH STACK OVERFLOW

```
df.isna()
```

	Unnamed: 0	id	age	salary	bought TV
0	False	False	False	False	False

```
from sklearn.preprocessing import Binarizer
df=pd.read_csv("//content//data (2).csv")
df
```

	Unnamed: 0	id	age	salary	bought TV
0	0	1	51	8241.0	1
1	1	2	33	68076.0	0
2	2	3	41	7876.0	0
3	3	4	60	94066.0	1
4	4	5	58	37654.0	0
...	...	...	...	...	...
395	395	396	50	61700.0	1
396	396	397	45	2630.0	1
397	397	398	53	68894.0	1
398	398	399	NaN	16818.0	0
399	399	400	47	8294.0	1

400 rows × 5 columns

```
df.head()
```

	Unnamed: 0	id	age	salary	bought TV
0	0	1	51	8241.0	1
1	1	2	33	68076.0	0
2	2	3	41	7876.0	0
3	3	4	60	94066.0	1
4	4	5	58	37654.0	0

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(df)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-35-b7d920039c80> in <module>()
      1 from sklearn.preprocessing import StandardScaler
      2 scaler = StandardScaler()
----> 3 scaler.fit(df)

 5 frames
/usr/local/lib/python3.7/dist-packages/numpy/core/_asarray.py in asarray(a, dtype, order)
     81
     82     """
---> 83     return array(a, dtype, copy=False, order=order)
     84
     85

ValueError: could not convert string to float: 'Null'
```

SEARCH STACK OVERFLOW

0s completed at 5:00 PM