In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
# To Partition the Data
from sklearn.model_selection import train_test_split
# Importing library for Logistic Regression
from sklearn.linear_model import LogisticRegression
# Importing  performance metrics - accuracy score and confusion matrix
from sklearn.metrics import accuracy_score,confusion_matrix
```

In [2]:
```
Data_income=pd.read_csv("C:/Users/shrey/Desktop/R PROGRAMMING(SIMULATION)/ir
Data_income.head(10)
```

Out[2]:

	age	JobType	EdType	maritalstatus	occupation	relationship	race	gender	capitalgain
0	45	Private	HS-grad	Divorced	Adm-clerical	Not-in-family	White	Female	0
1	24	Federal-gov	HS-grad	Never-married	Armed-Forces	Own-child	White	Male	0
2	44	Private	Some-college	Married-civ-spouse	Prof-specialty	Husband	White	Male	0
3	27	Private	9th	Never-married	Craft-repair	Other-relative	White	Male	0
4	20	Private	Some-college	Never-married	Sales	Not-in-family	White	Male	0
5	44	Private	HS-grad	Widowed	Exec-managerial	Unmarried	Black	Female	0
6	51	Private	HS-grad	Married-civ-spouse	Craft-repair	Husband	Amer-Indian-Eskimo	Male	0
7	20	Private	HS-grad	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0
8	17	?	11th	Never-married	?	Own-child	White	Female	0
9	19	Private	HS-grad	Never-married	Machine-op-inspct	Own-child	Black	Female	0

In [3]:
```
income=Data_income.copy()
income
```

Out[3]:

	age	JobType	EdType	maritalstatus	occupation	relationship	race	gender	capita
0	45	Private	HS-grad	Divorced	Adm-clerical	Not-in-family	White	Female	
1	24	Federal-gov	HS-grad	Never-married	Armed-Forces	Own-child	White	Male	
2	44	Private	Some-college	Married-civ-spouse	Prof-specialty	Husband	White	Male	
3	27	Private	9th	Never-married	Craft-repair	Other-relative	White	Male	
4	20	Private	Some-college	Never-married	Sales	Not-in-family	White	Male	
...	...	...	...	...	...	...	...	...	...
31973	34	Local-gov	HS-grad	Never-married	Farming-fishing	Not-in-family	Black	Male	
31974	34	Local-gov	Some-college	Never-married	Protective-serv	Not-in-family	White	Female	
31975	23	Private	Some-college	Married-civ-spouse	Adm-clerical	Husband	White	Male	
31976	42	Local-gov	Some-college	Married-civ-spouse	Adm-clerical	Wife	White	Female	
31977	29	Private	Bachelors	Never-married	Prof-specialty	Not-in-family	White	Male	

31978 rows × 13 columns

In [4]:
```python
print(income.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31978 entries, 0 to 31977
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   age            31978 non-null  int64 
 1   JobType        31978 non-null  object
 2   EdType         31978 non-null  object
 3   maritalstatus  31978 non-null  object
 4   occupation     31978 non-null  object
 5   relationship   31978 non-null  object
 6   race           31978 non-null  object
 7   gender         31978 non-null  object
 8   capitalgain    31978 non-null  int64 
 9   capitalloss    31978 non-null  int64 
 10  hoursperweek   31978 non-null  int64 
 11  nativecountry  31978 non-null  object
 12  SalStat        31978 non-null  object
dtypes: int64(4), object(9)
memory usage: 3.2+ MB
None
```

In [5]:
```python
## Check for missing values
income.isnull()
print("Data columns with null values :\n",income.isnull().sum())
```

```
Data columns with null values :
 age              0
JobType          0
EdType           0
maritalstatus    0
occupation       0
relationship     0
race             0
gender           0
capitalgain      0
capitalloss      0
hoursperweek     0
nativecountry    0
SalStat          0
dtype: int64
```

In [6]:
```python
# Summary of Numerical Variables
summary_num=income.describe()
print(summary_num)
```

```
                age   capitalgain   capitalloss  hoursperweek
count  31978.000000  31978.000000  31978.000000  31978.000000
mean      38.579023   1064.360623     86.739352     40.417850
std       13.662085   7298.596271    401.594301     12.345285
min       17.000000      0.000000      0.000000      1.000000
25%       28.000000      0.000000      0.000000     40.000000
50%       37.000000      0.000000      0.000000     40.000000
75%       48.000000      0.000000      0.000000     45.000000
max       90.000000  99999.000000   4356.000000     99.000000
```

In [7]:
```python
# Summary of Categorical Variables
summary_cate=income.describe(include="object")
print(summary_cate)
```

```
          JobType     EdType        maritalstatus        occupation relations
hip  \
count       31978      31978                31978             31978        31
978
unique          9         16                    7                15
6
top       Private    HS-grad   Married-civ-spouse    Prof-specialty      Husb
and
freq        22286      10368                14692              4038        12
947

          race gender  nativecountry                          SalStat
count    31978  31978          31978                            31978
unique       5      2             41                                2
top      White   Male  United-States   less than or equal to 50,000
freq     27430  21370          29170                            24283
```

In [8]:
```python
# Frequency of each categories
income["JobType"].value_counts()
```

Out[8]:
```
 Private             22286
 Self-emp-not-inc     2499
 Local-gov            2067
 ?                    1809
 State-gov            1279
 Self-emp-inc         1074
 Federal-gov           943
 Without-pay            14
 Never-worked            7
Name: JobType, dtype: int64
```

In [9]:
```python
income["occupation"].value_counts()
```

Out[9]:
```
 Prof-specialty       4038
 Craft-repair         4030
 Exec-managerial      3992
 Adm-clerical         3721
 Sales                3584
 Other-service        3212
 Machine-op-inspct    1966
 ?                    1816
 Transport-moving     1572
 Handlers-cleaners    1350
 Farming-fishing       989
 Tech-support          912
 Protective-serv       644
 Priv-house-serv       143
 Armed-Forces            9
Name: occupation, dtype: int64
```

In [10]:
```python
# Checking for unique classes
print(np.unique(income["JobType"]))
```

```
[' ?' ' Federal-gov' ' Local-gov' ' Never-worked' ' Private'
 ' Self-emp-inc' ' Self-emp-not-inc' ' State-gov' ' Without-pay']
```

In [11]:
```
print(np.unique(income["occupation"]))
```

```
[' ?' ' Adm-clerical' ' Armed-Forces' ' Craft-repair' ' Exec-managerial'
 ' Farming-fishing' ' Handlers-cleaners' ' Machine-op-inspct'
 ' Other-service' ' Priv-house-serv' ' Prof-specialty' ' Protective-serv'
 ' Sales' ' Tech-support' ' Transport-moving']
```

In [12]:
```
income=pd.read_csv("C:/Users/shrey/Desktop/R PROGRAMMING(SIMULATION)/income.
income
```

Out[12]:

	age	JobType	EdType	maritalstatus	occupation	relationship	race	gender	capita
0	45	Private	HS-grad	Divorced	Adm-clerical	Not-in-family	White	Female	
1	24	Federal-gov	HS-grad	Never-married	Armed-Forces	Own-child	White	Male	
2	44	Private	Some-college	Married-civ-spouse	Prof-specialty	Husband	White	Male	
3	27	Private	9th	Never-married	Craft-repair	Other-relative	White	Male	
4	20	Private	Some-college	Never-married	Sales	Not-in-family	White	Male	
...	...	...	...	...	...	...	...	...	...
31973	34	Local-gov	HS-grad	Never-married	Farming-fishing	Not-in-family	Black	Male	
31974	34	Local-gov	Some-college	Never-married	Protective-serv	Not-in-family	White	Female	
31975	23	Private	Some-college	Married-civ-spouse	Adm-clerical	Husband	White	Male	
31976	42	Local-gov	Some-college	Married-civ-spouse	Adm-clerical	Wife	White	Female	
31977	29	Private	Bachelors	Never-married	Prof-specialty	Not-in-family	White	Male	

31978 rows × 13 columns

Data Pre-Processing

In [13]:
```
income.isnull().sum()
```

Out[13]:
```
age                 0
JobType          1809
EdType              0
maritalstatus       0
occupation       1816
relationship        0
race                0
gender              0
capitalgain         0
capitalloss         0
hoursperweek        0
nativecountry       0
SalStat             0
dtype: int64
```

In [14]:
```
missing=income[income.isnull().any(axis=1)]
missing
```

Out[14]:

	age	JobType	EdType	maritalstatus	occupation	relationship	race	gender	capita
8	17	NaN	11th	Never-married	NaN	Own-child	White	Female	
17	32	NaN	Some-college	Married-civ-spouse	NaN	Husband	White	Male	
29	22	NaN	Some-college	Never-married	NaN	Own-child	White	Male	
42	52	NaN	12th	Never-married	NaN	Other-relative	Black	Male	
44	63	NaN	1st-4th	Married-civ-spouse	NaN	Husband	White	Male	
...	...	...	...	...	...	...	...	...	
31892	59	NaN	Bachelors	Married-civ-spouse	NaN	Husband	White	Male	
31934	20	NaN	HS-grad	Never-married	NaN	Other-relative	White	Female	
31945	28	NaN	Some-college	Married-civ-spouse	NaN	Wife	White	Female	
31967	80	NaN	HS-grad	Widowed	NaN	Not-in-family	White	Male	
31968	17	NaN	11th	Never-married	NaN	Own-child	White	Male	

1816 rows × 13 columns

In [15]:
```
income_2=income.dropna(axis=0)
income_2
```

Out[15]:

	age	JobType	EdType	maritalstatus	occupation	relationship	race	gender	capita
0	45	Private	HS-grad	Divorced	Adm-clerical	Not-in-family	White	Female	
1	24	Federal-gov	HS-grad	Never-married	Armed-Forces	Own-child	White	Male	
2	44	Private	Some-college	Married-civ-spouse	Prof-specialty	Husband	White	Male	
3	27	Private	9th	Never-married	Craft-repair	Other-relative	White	Male	
4	20	Private	Some-college	Never-married	Sales	Not-in-family	White	Male	
...	...	...	...	...	...	...	...	...	...
31973	34	Local-gov	HS-grad	Never-married	Farming-fishing	Not-in-family	Black	Male	
31974	34	Local-gov	Some-college	Never-married	Protective-serv	Not-in-family	White	Female	
31975	23	Private	Some-college	Married-civ-spouse	Adm-clerical	Husband	White	Male	
31976	42	Local-gov	Some-college	Married-civ-spouse	Adm-clerical	Wife	White	Female	
31977	29	Private	Bachelors	Never-married	Prof-specialty	Not-in-family	White	Male	

30162 rows × 13 columns

In [16]:
```
# Relationships between independent variables
correlation=income_2.corr()
correlation
```

Out[16]:

	age	capitalgain	capitalloss	hoursperweek
age	1.000000	0.080154	0.060165	0.101599
capitalgain	0.080154	1.000000	-0.032229	0.080432
capitalloss	0.060165	-0.032229	1.000000	0.052417
hoursperweek	0.101599	0.080432	0.052417	1.000000

Cross Tables and Data Visualisation

In [17]:
```
# Extracting the column names
income_2.columns
```

Out[17]:
```
Index(['age', 'JobType', 'EdType', 'maritalstatus', 'occupation',
       'relationship', 'race', 'gender', 'capitalgain', 'capitalloss',
       'hoursperweek', 'nativecountry', 'SalStat'],
      dtype='object')
```

In [18]:
```
# Gender Proportion Table
gender=pd.crosstab(index=income_2["gender"],columns="count",normalize=True)
print(gender)
```

```
col_0       count
gender
 Female  0.324315
 Male    0.675685
```

Gender vs Salary

In [19]:
```
gender_salstat=pd.crosstab(index=income_2["gender"],columns=income_2["SalSta
print(gender_salstat)
```

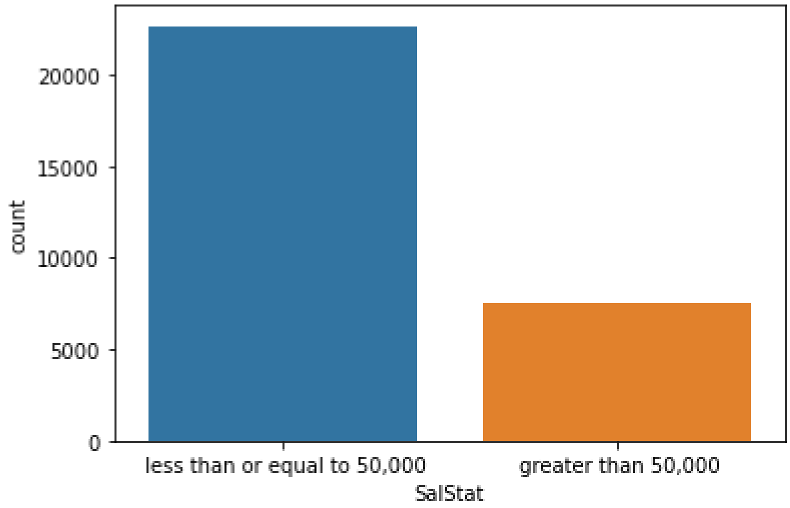
```
SalStat   greater than 50,000   less than or equal to 50,000
gender
 Female              0.113678                       0.886322
 Male                0.313837                       0.686163
All                  0.248922                       0.751078
```

Frequency Distribution of the Salary Status

```
In [20]: salstat=sns.countplot(income_2["SalStat"])
         salstat
```

```
C:\Users\shrey\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

```
Out[20]: <AxesSubplot:xlabel='SalStat', ylabel='count'>
```

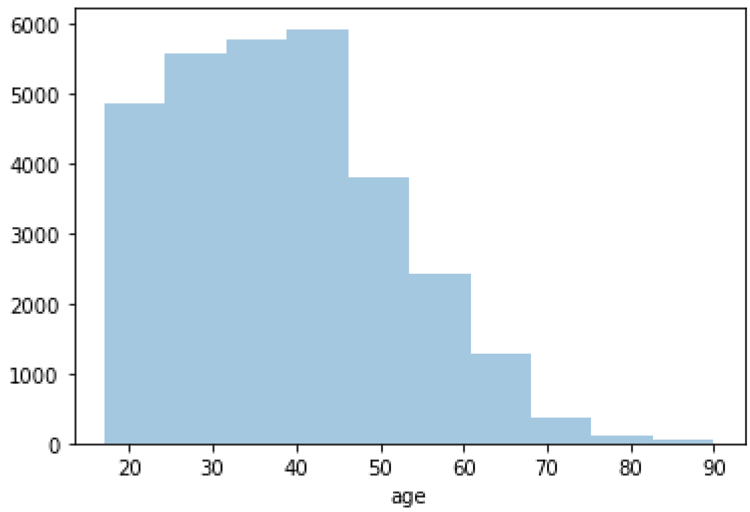


Histogram of Age

```
In [21]: sns.distplot(income_2["age"],bins=10,kde=False)
```

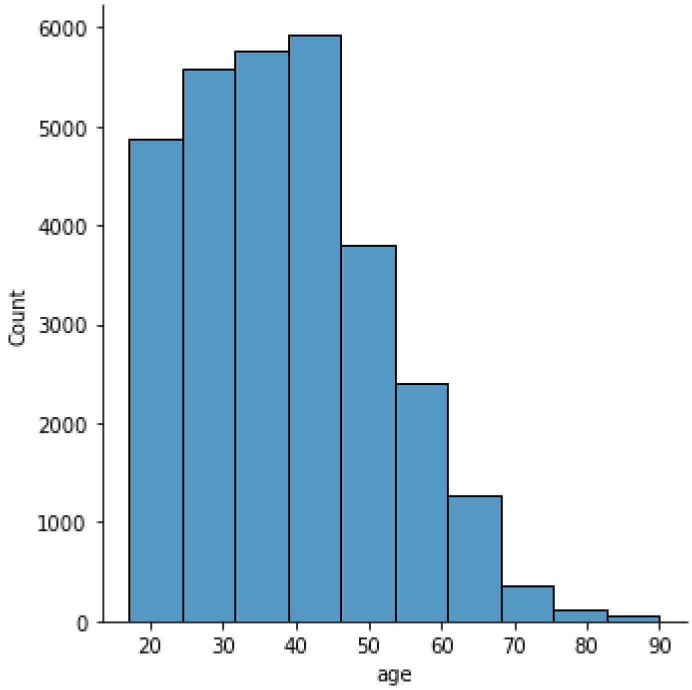
```
C:\Users\shrey\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

```
Out[21]: <AxesSubplot:xlabel='age'>
```



In [22]:
```
sns.displot(income_2["age"],bins=10,kde=False)
```

Out[22]: <seaborn.axisgrid.FacetGrid at 0x16e9d164610>

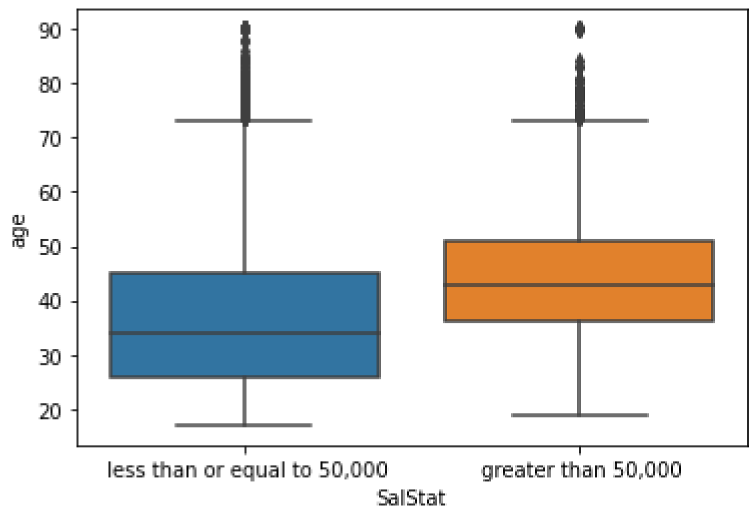


Boxplot - Age vs Salary Status

In [23]:
```
sns.boxplot("SalStat","age",data=income_2)
```

```
C:\Users\shrey\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misint
erpretation.
  warnings.warn(
```

Out[23]: <AxesSubplot:xlabel='SalStat', ylabel='age'>



In [24]:
```python
income_2.groupby("SalStat")["age"].median()
```

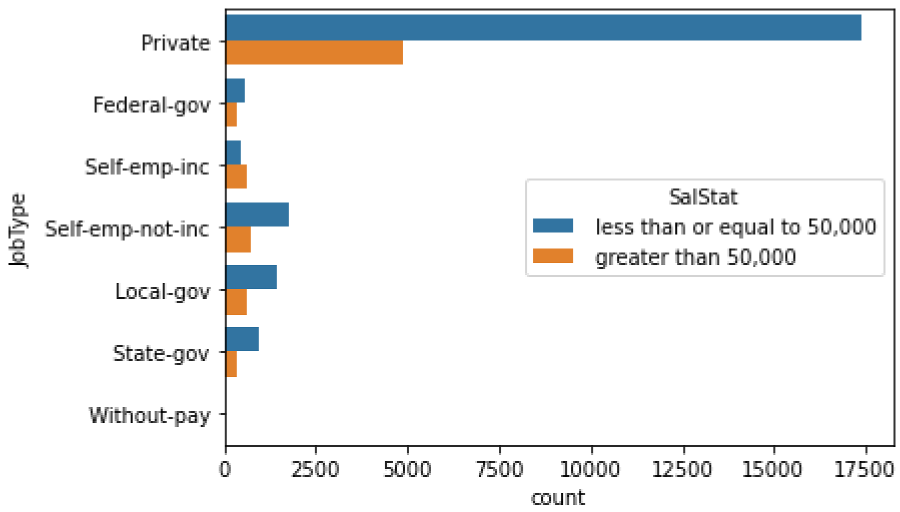
Out[24]:
```
SalStat
 greater than 50,000             43.0
 less than or equal to 50,000    34.0
Name: age, dtype: float64
```

In [25]:
```python
# People with age 35-40 are likely to earn > 50000
# People with age 25-35 are likely to earn <= 50000
```

Exploratory Data Analysis

In [28]:
```python
# JobType Vs Salary Status
sns.countplot(y='JobType',hue='SalStat',data=income_2)
```

Out[28]: <AxesSubplot:xlabel='count', ylabel='JobType'>



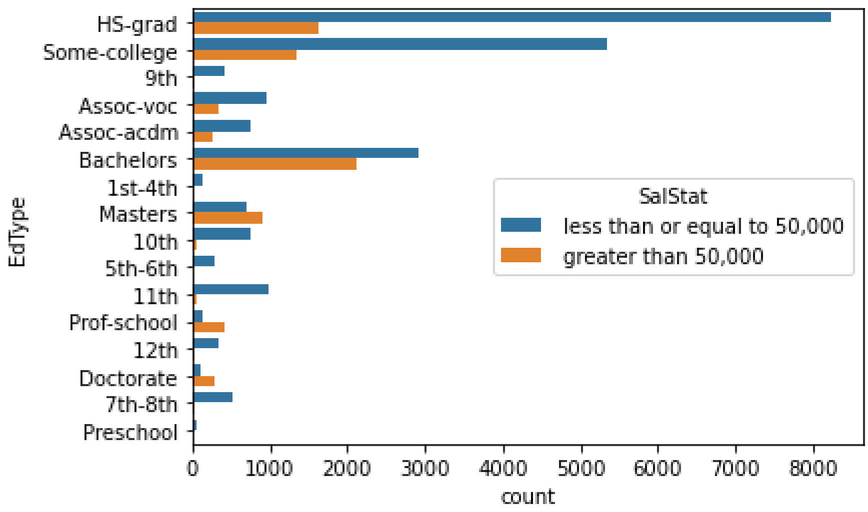
In [29]:
```python
JobType_salstat=pd.crosstab(index=income_2["JobType"],columns=income_2["SalS
print(JobType_salstat)
```

```
SalStat             greater than 50,000   less than or equal to 50,000
JobType
 Federal-gov                   0.387063                       0.612937
 Local-gov                     0.294630                       0.705370
 Private                       0.218792                       0.781208
 Self-emp-inc                  0.558659                       0.441341
 Self-emp-not-inc              0.285714                       0.714286
 State-gov                     0.268960                       0.731040
 Without-pay                   0.000000                       1.000000
All                            0.248922                       0.751078
```

In [30]:
```python
# From the above table 56% of the self-employed people earn more than 50000
```

In [32]:
```
# Education Vs Salary Status
sns.countplot(y='EdType',hue='SalStat',data=income_2)
```

Out[32]: <AxesSubplot:xlabel='count', ylabel='EdType'>



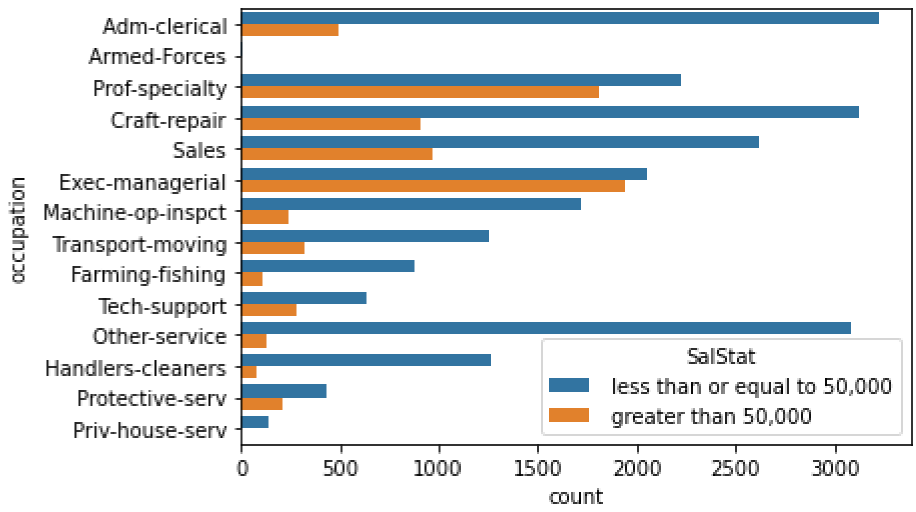
In [33]:
```
EdType_salstat=pd.crosstab(index=income_2["EdType"],columns=income_2["SalSta
print(EdType_salstat)
```

```
SalStat         greater than 50,000  less than or equal to 50,000
EdType
 10th                      0.071951                      0.928049
 11th                      0.056298                      0.943702
 12th                      0.076923                      0.923077
 1st-4th                   0.039735                      0.960265
 5th-6th                   0.041667                      0.958333
 7th-8th                   0.062837                      0.937163
 9th                       0.054945                      0.945055
 Assoc-acdm                0.253968                      0.746032
 Assoc-voc                 0.263198                      0.736802
 Bachelors                 0.421491                      0.578509
 Doctorate                 0.746667                      0.253333
 HS-grad                   0.164329                      0.835671
 Masters                   0.564229                      0.435771
 Preschool                 0.000000                      1.000000
 Prof-school               0.749077                      0.250923
 Some-college              0.200060                      0.799940
All                        0.248922                      0.751078
```

In [ ]:
```
# From the above table we can see that who have done Doctorates,Masters and
```

In [34]:
```
# Occupation Vs Salary Status
sns.countplot(y='occupation',hue='SalStat',data=income_2)
```

Out[34]: <AxesSubplot:xlabel='count', ylabel='occupation'>



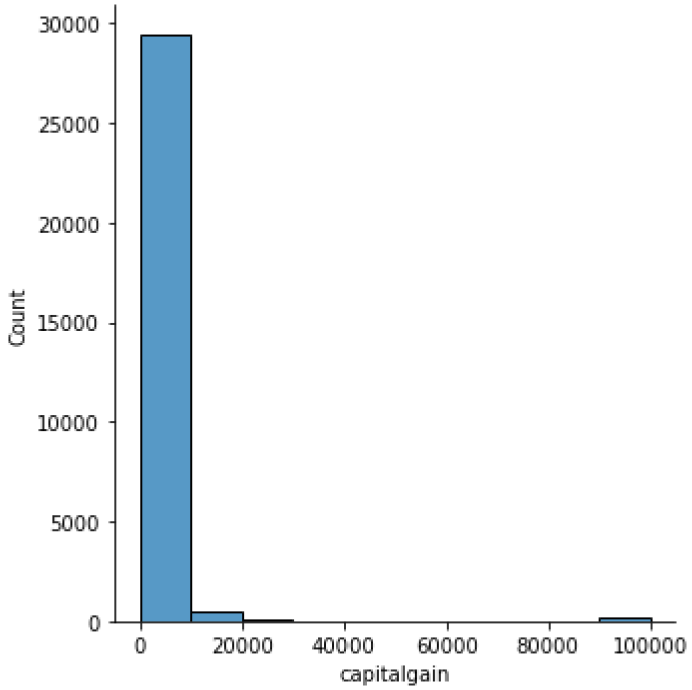
In [35]:
```
Occupation_salstat=pd.crosstab(index=income_2["occupation"],columns=income_2
print(Occupation_salstat)
```

```
SalStat             greater than 50,000   less than or equal to 50,000
occupation
 Adm-clerical                  0.133835                       0.866165
 Armed-Forces                  0.111111                       0.888889
 Craft-repair                  0.225310                       0.774690
 Exec-managerial               0.485220                       0.514780
 Farming-fishing               0.116279                       0.883721
 Handlers-cleaners             0.061481                       0.938519
 Machine-op-inspct             0.124619                       0.875381
 Other-service                 0.041096                       0.958904
 Priv-house-serv               0.006993                       0.993007
 Prof-specialty                0.448489                       0.551511
 Protective-serv               0.326087                       0.673913
 Sales                         0.270647                       0.729353
 Tech-support                  0.304825                       0.695175
 Transport-moving              0.202926                       0.797074
All                            0.248922                       0.751078
```

In [36]:
```
# Those who make more than 50000 USD per year are more likely to work as man
```

In [42]:
```
sns.displot(income_2["capitalgain"],bins=10,kde=False)
```

Out[42]: <seaborn.axisgrid.FacetGrid at 0x16ea2e26970>



In [43]:
```
capitalgain=pd.crosstab(index=income_2["capitalgain"],columns="count",normal
print(capitalgain)
```

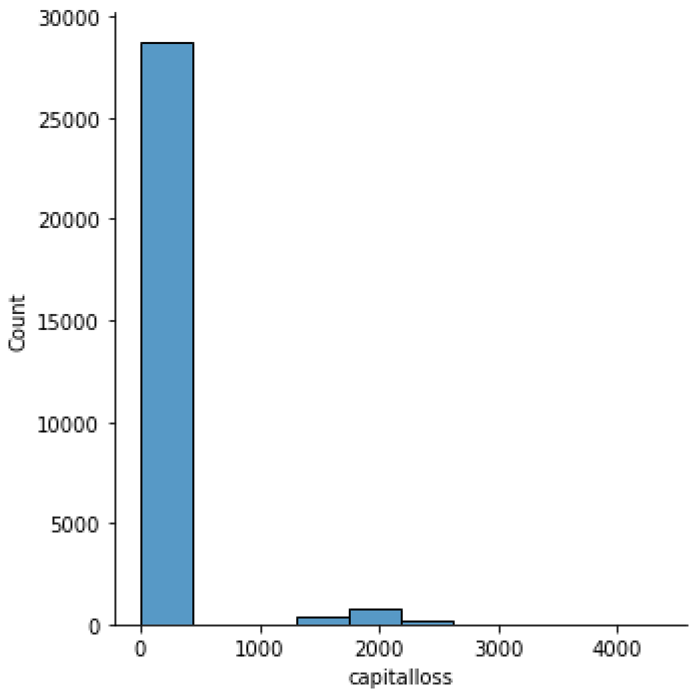
```
col_0          count
capitalgain
0           0.915854
114         0.000199
401         0.000033
594         0.000928
914         0.000265
...              ...
25236       0.000365
27828       0.001061
34095       0.000099
41310       0.000066
99999       0.004907

[118 rows x 1 columns]
```

In [ ]:
```
# 92% of the capital gain is zero
```

In [41]:
```
sns.displot(income_2["capitalloss"],bins=10,kde=False)
```

Out[41]: <seaborn.axisgrid.FacetGrid at 0x16e9fcb6460>



In [44]:
```
capitalloss=pd.crosstab(index=income_2["capitalloss"],columns="count",normal
print(capitalloss)
```

```
col_0          count
capitalloss
0           0.952689
155         0.000033
213         0.000133
323         0.000099
419         0.000033
...              ...
3004        0.000033
3683        0.000066
3770        0.000066
3900        0.000066
4356        0.000033

[90 rows x 1 columns]
```

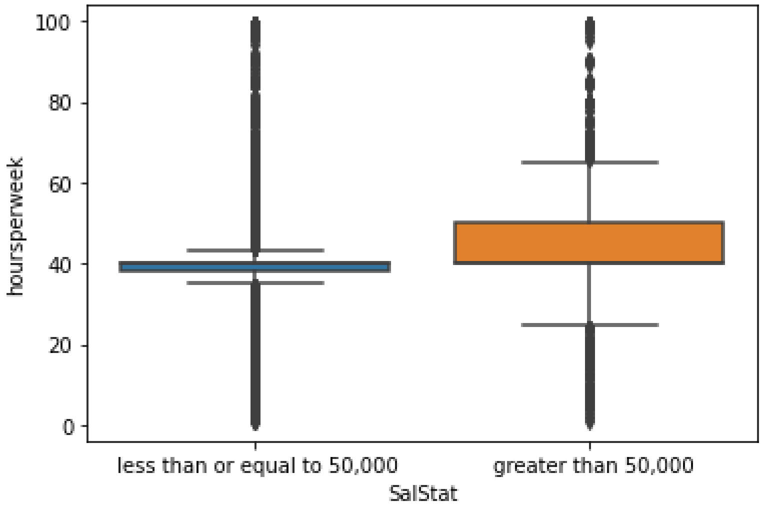
In [ ]:
```
# 95% of the capital loss is zero
```

Hours per week vs Salary Status

In [51]:
```
sns.boxplot("SalStat","hoursperweek",data=income_2)
```

```
C:\Users\shrey\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[51]: `<AxesSubplot:xlabel='SalStat', ylabel='hoursperweek'>`



In [49]:
```
# From the above plot it is very clear that those who earn more than 50000 a
```

In [ ]:

In [ ]: