In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.model_selection import GridSearchCV
```

In [2]:

```
train=pd.read_csv("C:\\Users\\Admin\\Downloads\\assignment 9\\SalaryData_Train(1).csv")
train
```

Out[2]:

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	sex
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female
...	...	...	...	...	...	...	...	...	...
30156	27	Private	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female
30157	40	Private	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male
30158	58	Private	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female
30159	22	Private	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male
30160	52	Self-emp-inc	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female

30161 rows × 14 columns

In [3]:

```
test=pd.read_csv("C:\\Users\\Admin\\Downloads\\assignment 9\\SalaryData_Test(1).csv")
test
```

Out[3]:

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	se
0	25	Private	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Mal

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	se
1	38	Private	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Mal
2	28	Local-gov	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Mal
3	44	Private	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Mal
4	34	Private	10th	6	Never-married	Other-service	Not-in-family	White	Mal
...	...	...	...	...	...	...	...	...	..
15055	33	Private	Bachelors	13	Never-married	Prof-specialty	Own-child	White	Mal
15056	39	Private	Bachelors	13	Divorced	Prof-specialty	Not-in-family	White	Femal
15057	38	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White	Mal
15058	44	Private	Bachelors	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander	Mal
15059	35	Self-emp-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Mal

15060 rows × 14 columns

In [4]:

```
train.head()
```

Out[4]:

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	sex	capi
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	

In [5]:

```
test.head()
```

Out[5]:

age	workclass	education	educationno	maritalstatus	occupation	relationship	race	sex	capita

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	sex	capita
0	25	Private	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	
1	38	Private	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	
2	28	Local-gov	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	
3	44	Private	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	
4	34	Private	10th	6	Never-married	Other-service	Not-in-family	White	Male	

In [7]:

```
train.shape
```

Out[7]:

```
(30161, 14)
```

In [8]:

```
test.shape
```

Out[8]:

```
(15060, 14)
```

In [9]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30161 entries, 0 to 30160
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   age            30161 non-null  int64 
 1   workclass      30161 non-null  object
 2   education      30161 non-null  object
 3   educationno    30161 non-null  int64 
 4   maritalstatus  30161 non-null  object
 5   occupation     30161 non-null  object
 6   relationship   30161 non-null  object
 7   race           30161 non-null  object
 8   sex            30161 non-null  object
 9   capitalgain    30161 non-null  int64 
 10  capitalloss    30161 non-null  int64 
 11  hoursperweek   30161 non-null  int64 
 12  native         30161 non-null  object
 13  Salary         30161 non-null  object
dtypes: int64(5), object(9)
memory usage: 3.2+ MB
```

In [10]:

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15060 entries, 0 to 15059
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype 
```

```
 ---  ------         --------------  -----
 0    age            15060 non-null  int64
 1    workclass      15060 non-null  object
 2    education      15060 non-null  object
 3    educationno    15060 non-null  int64
 4    maritalstatus  15060 non-null  object
 5    occupation     15060 non-null  object
 6    relationship   15060 non-null  object
 7    race           15060 non-null  object
 8    sex            15060 non-null  object
 9    capitalgain    15060 non-null  int64
 10   capitalloss    15060 non-null  int64
 11   hoursperweek   15060 non-null  int64
 12   native         15060 non-null  object
 13   Salary         15060 non-null  object
dtypes: int64(5), object(9)
memory usage: 1.6+ MB
```

In [11]:

```
train.describe()
```

Out[11]:

	age	educationno	capitalgain	capitalloss	hoursperweek
count	30161.000000	30161.000000	30161.000000	30161.000000	30161.000000
mean	38.438115	10.121316	1092.044064	88.302311	40.931269
std	13.134830	2.550037	7406.466611	404.121321	11.980182
min	17.000000	1.000000	0.000000	0.000000	1.000000
25%	28.000000	9.000000	0.000000	0.000000	40.000000
50%	37.000000	10.000000	0.000000	0.000000	40.000000
75%	47.000000	13.000000	0.000000	0.000000	45.000000
max	90.000000	16.000000	99999.000000	4356.000000	99.000000

In [12]:

```
test.describe()
```

Out[12]:

	age	educationno	capitalgain	capitalloss	hoursperweek
count	15060.000000	15060.000000	15060.000000	15060.000000	15060.000000
mean	38.768327	10.112749	1120.301594	89.041899	40.951594
std	13.380676	2.558727	7703.181842	406.283245	12.062831
min	17.000000	1.000000	0.000000	0.000000	1.000000
25%	28.000000	9.000000	0.000000	0.000000	40.000000
50%	37.000000	10.000000	0.000000	0.000000	40.000000
75%	48.000000	13.000000	0.000000	0.000000	45.000000
max	90.000000	16.000000	99999.000000	3770.000000	99.000000

In [13]:

```
train.corr()
```

Out[13]:

	age	educationno	capitalgain	capitalloss	hoursperweek
age	1.000000	0.043525	0.080152	0.060278	0.101598
educationno	0.043525	1.000000	0.124416	0.079691	0.152522
capitalgain	0.080152	0.124416	1.000000	-0.032218	0.080431
capitalloss	0.060278	0.079691	-0.032218	1.000000	0.052454
hoursperweek	0.101598	0.152522	0.080431	0.052454	1.000000

In [14]:

```
test.corr()
```

Out[14]:

	age	educationno	capitalgain	capitalloss	hoursperweek
age	1.000000	0.026123	0.078760	0.057745	0.102758
educationno	0.026123	1.000000	0.131750	0.085817	0.133691
capitalgain	0.078760	0.131750	1.000000	-0.031876	0.090501
capitalloss	0.057745	0.085817	-0.031876	1.000000	0.057712
hoursperweek	0.102758	0.133691	0.090501	0.057712	1.000000

In [15]:

```
lb = LabelEncoder()
```

In [16]:

```
train["workclass"] = lb.fit_transform(train["workclass"])
train["education"] = lb.fit_transform(train["education"])
train["maritalstatus"] = lb.fit_transform(train["maritalstatus"])
train["occupation"] = lb.fit_transform(train["occupation"])
train["relationship"] = lb.fit_transform(train["relationship"])
train["race"] = lb.fit_transform(train["race"])
train["sex"] = lb.fit_transform(train["sex"])
train["native"] = lb.fit_transform(train["native"])
train["Salary"] = lb.fit_transform(train["Salary"])
```

In [17]:

```
test["workclass"] = lb.fit_transform(test["workclass"])
test["education"] = lb.fit_transform(test["education"])
test["maritalstatus"] = lb.fit_transform(test["maritalstatus"])
test["occupation"] = lb.fit_transform(test["occupation"])
test["relationship"] = lb.fit_transform(test["relationship"])
test["race"] = lb.fit_transform(test["race"])
test["sex"] = lb.fit_transform(test["sex"])
test["native"] = lb.fit_transform(test["native"])
test["Salary"] = lb.fit_transform(test["Salary"])
```

In [18]:

```
train = train.iloc[: 2000, :]
```

In [19]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   age            2000 non-null   int64
 1   workclass      2000 non-null   int32
 2   education      2000 non-null   int32
 3   educationno    2000 non-null   int64
 4   maritalstatus  2000 non-null   int32
 5   occupation     2000 non-null   int32
 6   relationship   2000 non-null   int32
 7   race           2000 non-null   int32
 8   sex            2000 non-null   int32
 9   capitalgain    2000 non-null   int64
 10  capitalloss    2000 non-null   int64
 11  hoursperweek   2000 non-null   int64
 12  native         2000 non-null   int32
 13  Salary         2000 non-null   int32
dtypes: int32(9), int64(5)
memory usage: 148.6 KB
```

In [20]:

```
test = test.iloc[: 1300, :]
```

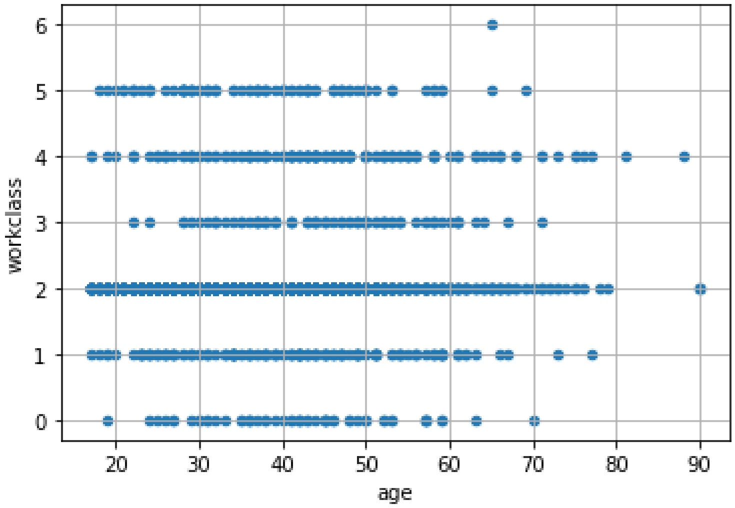
In [21]:

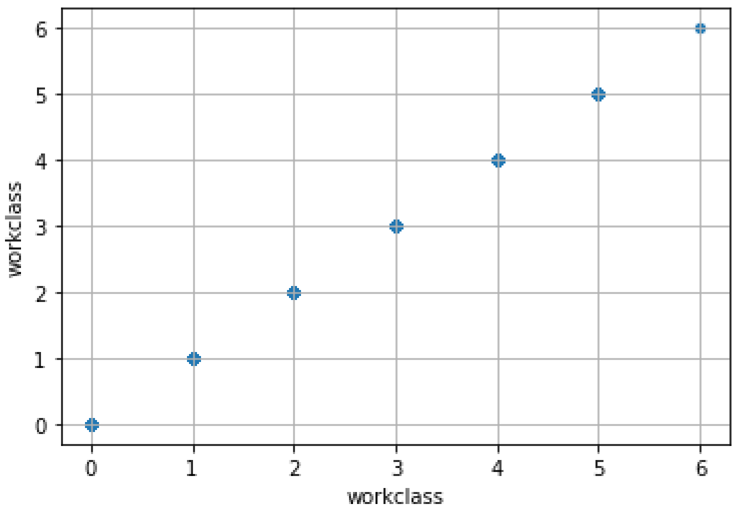
```
test.info()
```

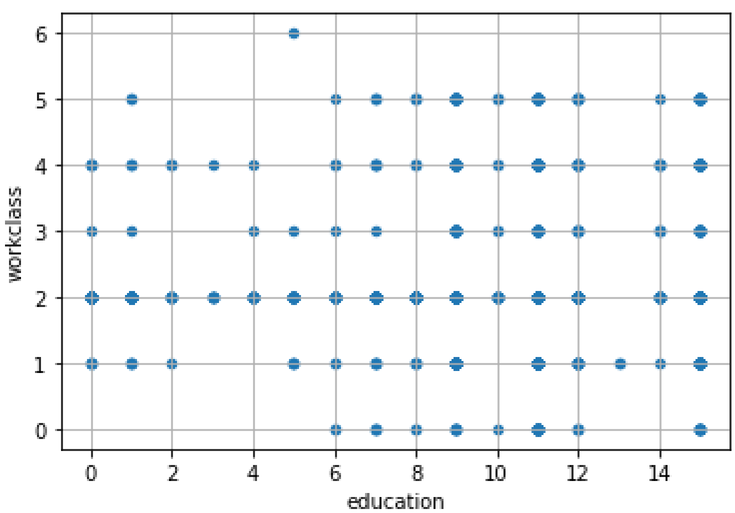
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1300 entries, 0 to 1299
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   age            1300 non-null   int64
 1   workclass      1300 non-null   int32
 2   education      1300 non-null   int32
 3   educationno    1300 non-null   int64
 4   maritalstatus  1300 non-null   int32
 5   occupation     1300 non-null   int32
 6   relationship   1300 non-null   int32
 7   race           1300 non-null   int32
 8   sex            1300 non-null   int32
 9   capitalgain    1300 non-null   int64
 10  capitalloss    1300 non-null   int64
 11  hoursperweek   1300 non-null   int64
 12  native         1300 non-null   int32
 13  Salary         1300 non-null   int32
dtypes: int32(9), int64(5)
memory usage: 96.6 KB
```

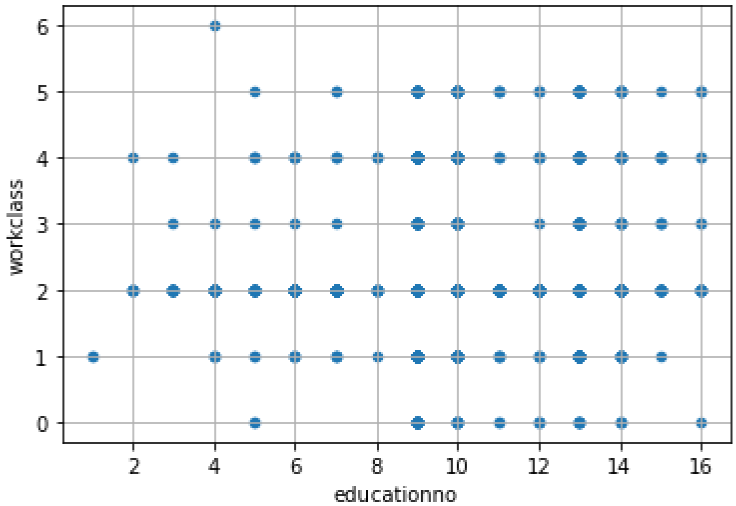
In [22]:

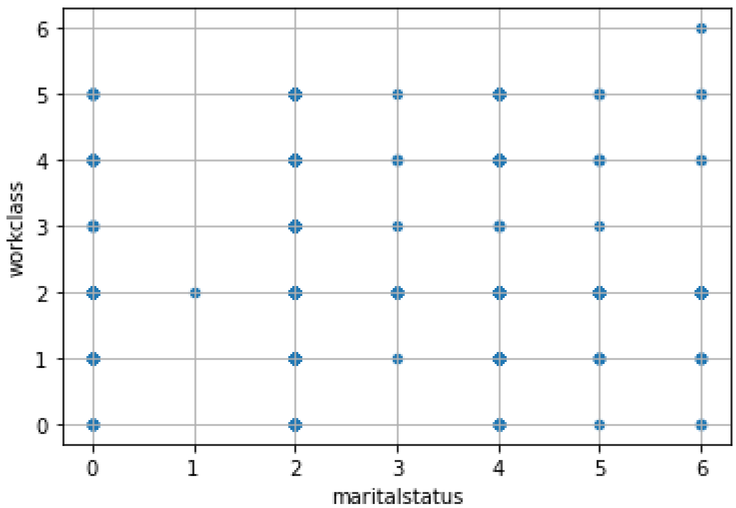
```
for i in train.describe().columns[:-2]:
    train.plot.scatter(i,'workclass',grid=True)
```

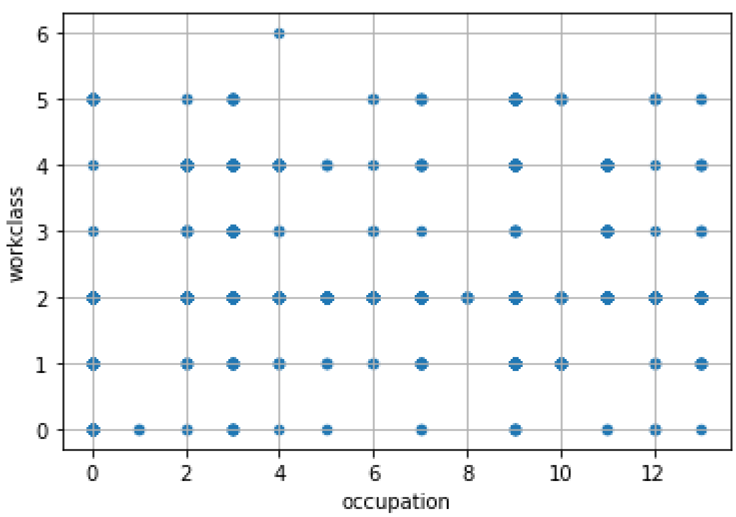

workclass
age
0
1
2
3
4
5
6
20
30
40
50
60
70
80
90

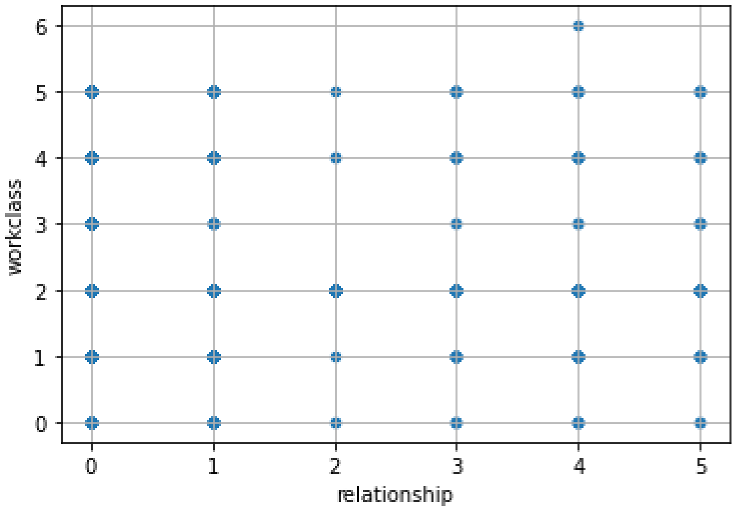

workclass
workclass
0
1
2
3
4
5
6

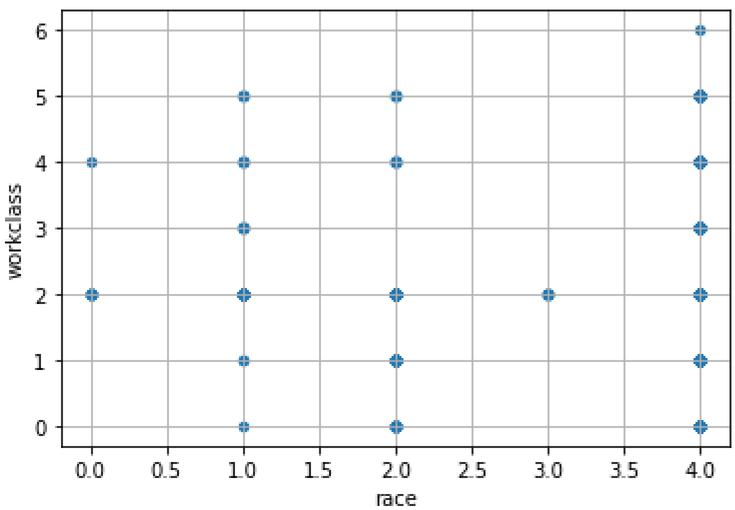

workclass
education
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14

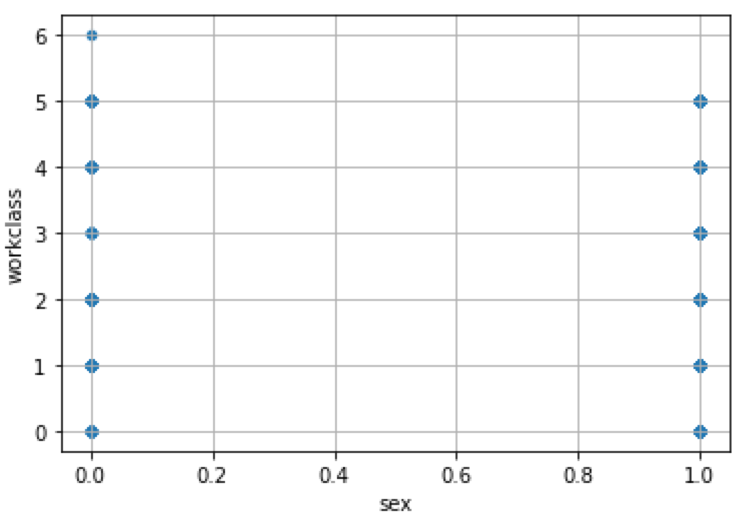

workclass
educationno

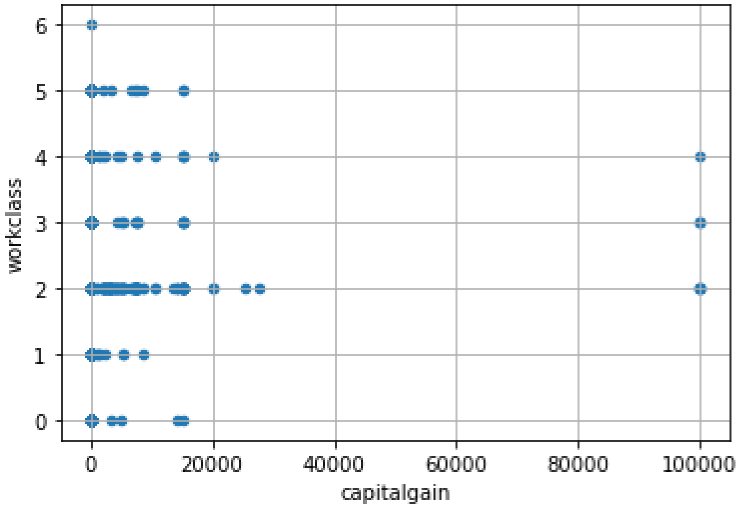

workclass
maritalstatus

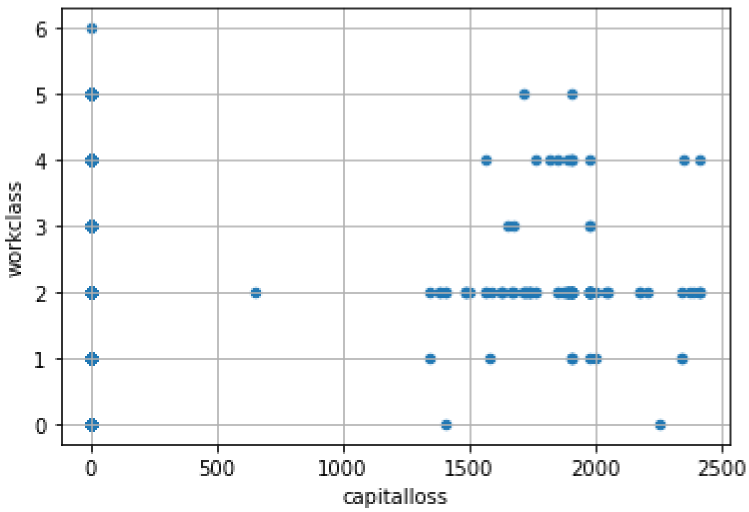

workclass
occupation

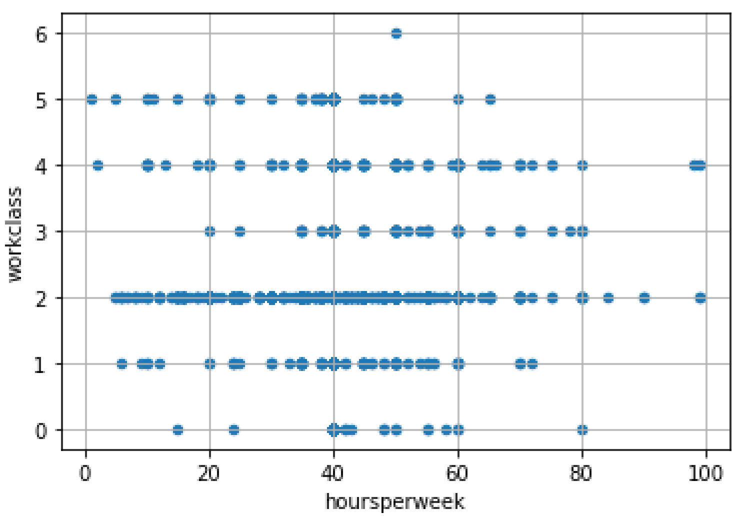

workclass
relationship
0
1
2
3
4
5
6


workclass
race
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


workclass
sex
0.0
0.2
0.4
0.6
0.8
1.0

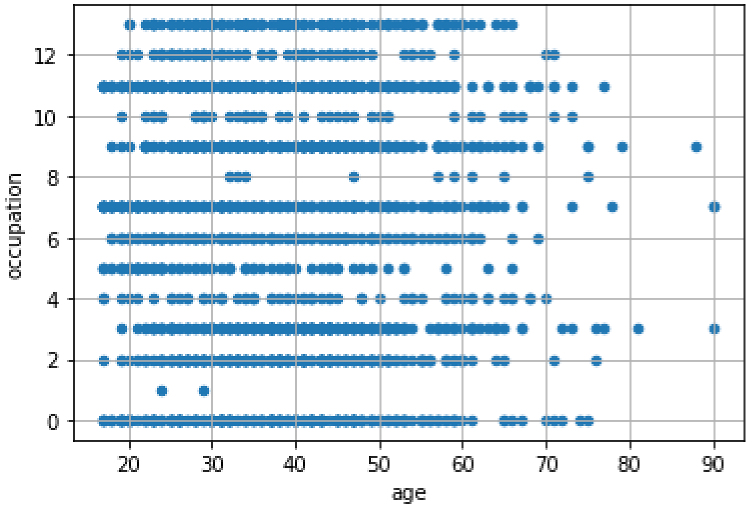


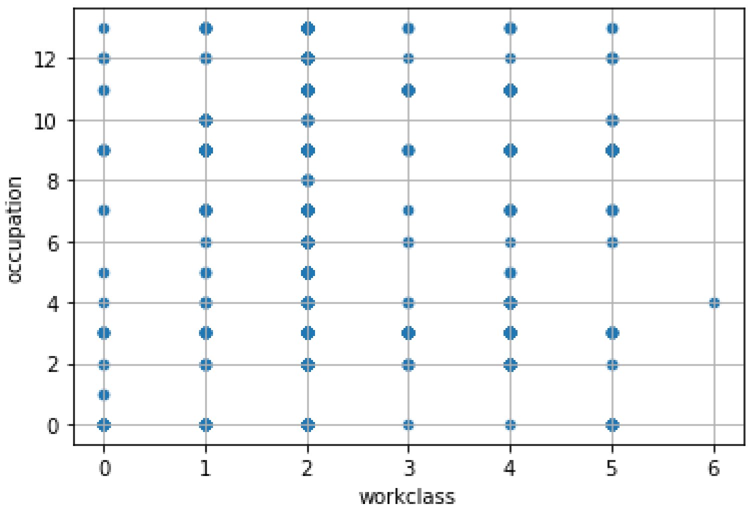


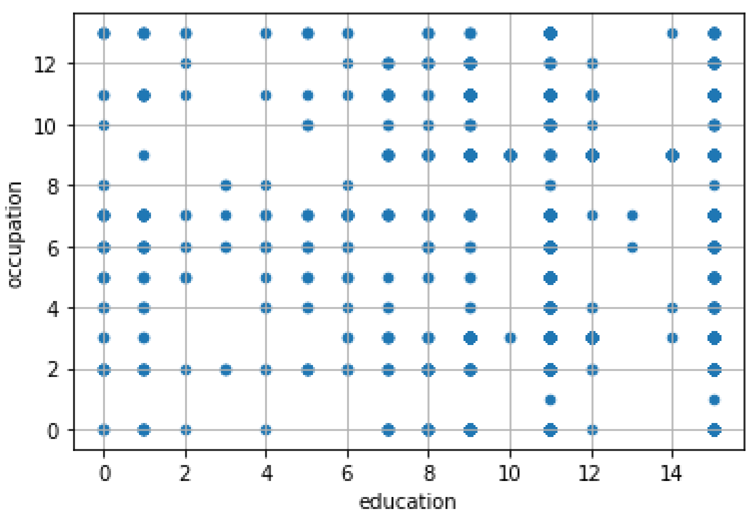


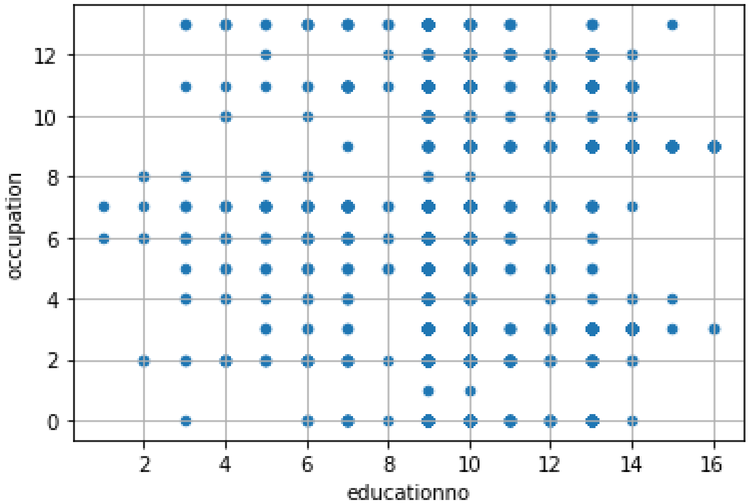
In [23]:

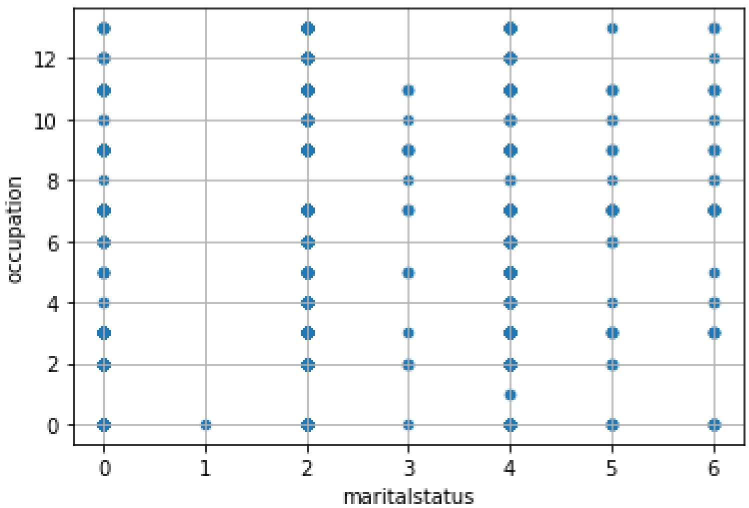
```
for i in train.describe().columns[:-2]:
    train.plot.scatter(i,'occupation',grid=True)
```

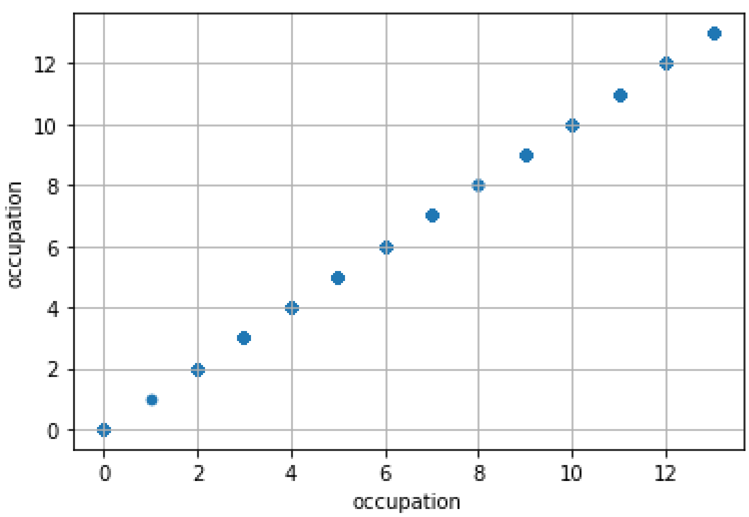

occupation
age

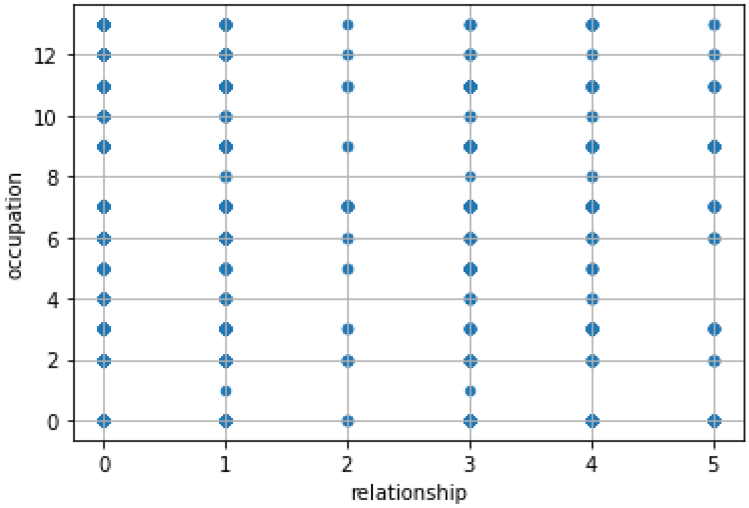

occupation
workclass

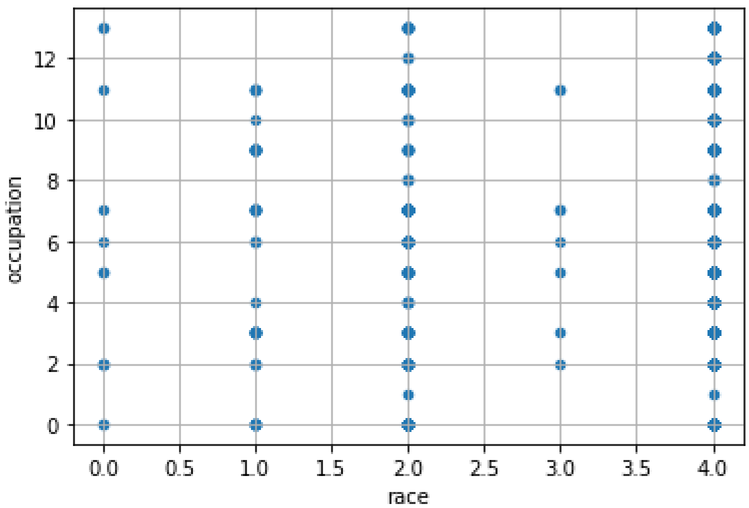

occupation
education

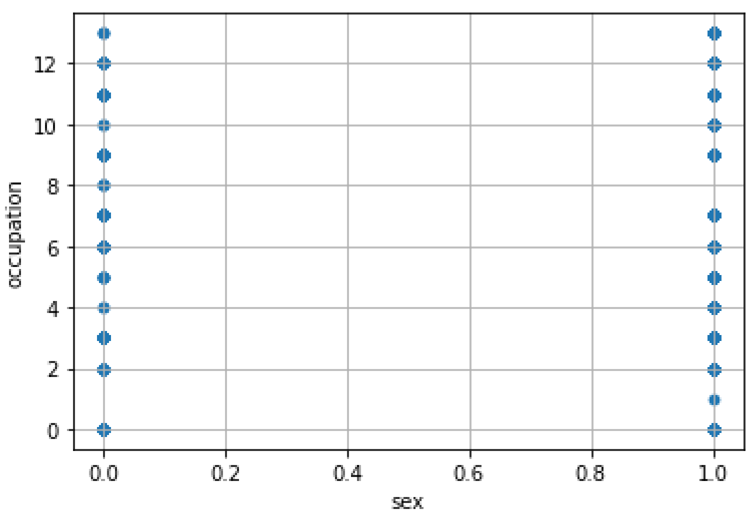

occupation
0
2
4
6
8
10
12
2
4
6
8
10
12
14
16
educationno

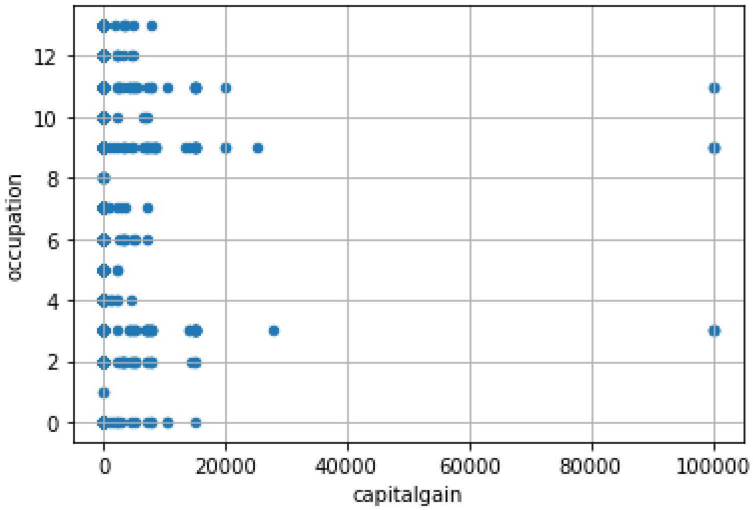

occupation
0
2
4
6
8
10
12
0
1
2
3
4
5
6
maritalstatus

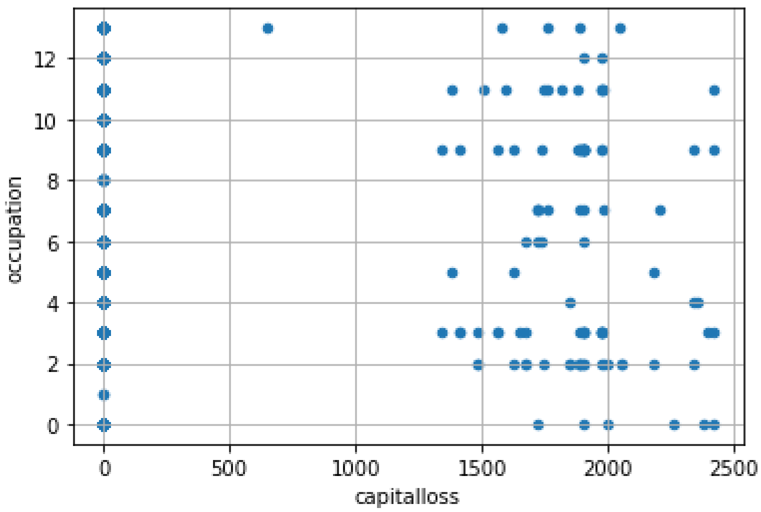

occupation
0
2
4
6
8
10
12
0
2
4
6
8
10
12
occupation

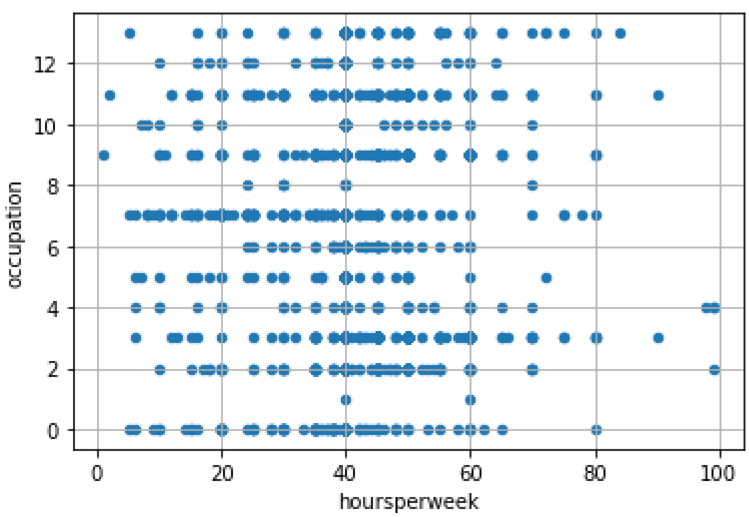

occupation
relationship
0
2
4
6
8
10
12
1
3
5


occupation
race
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


occupation
sex
0.0
0.2
0.4
0.6
0.8
1.0

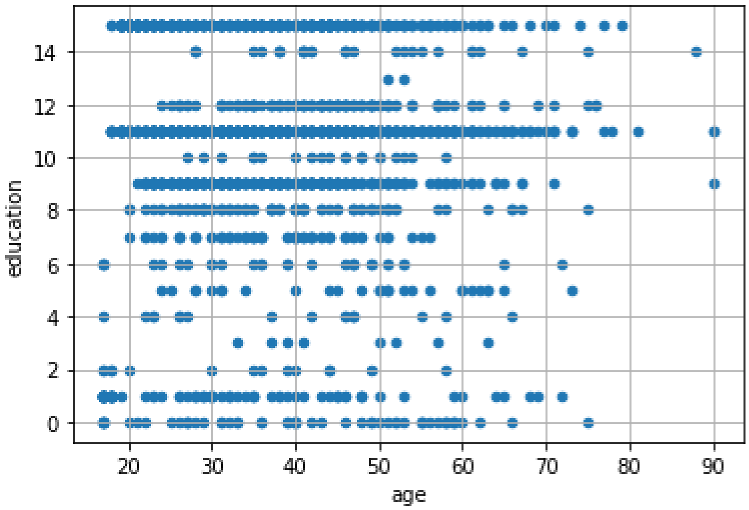


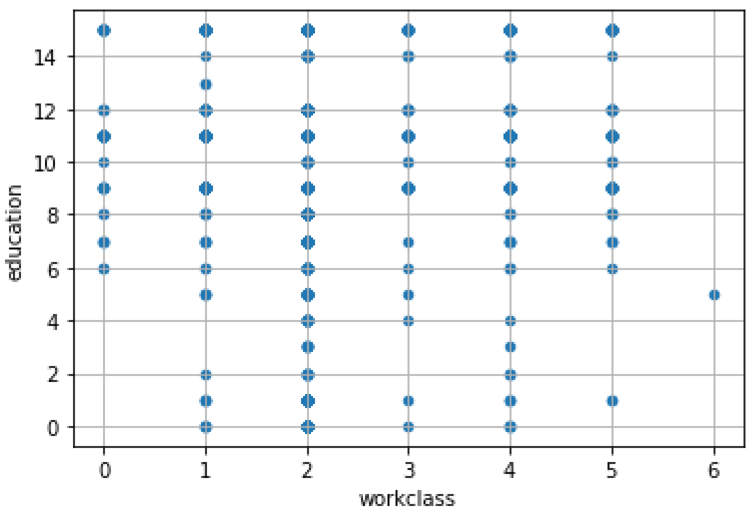


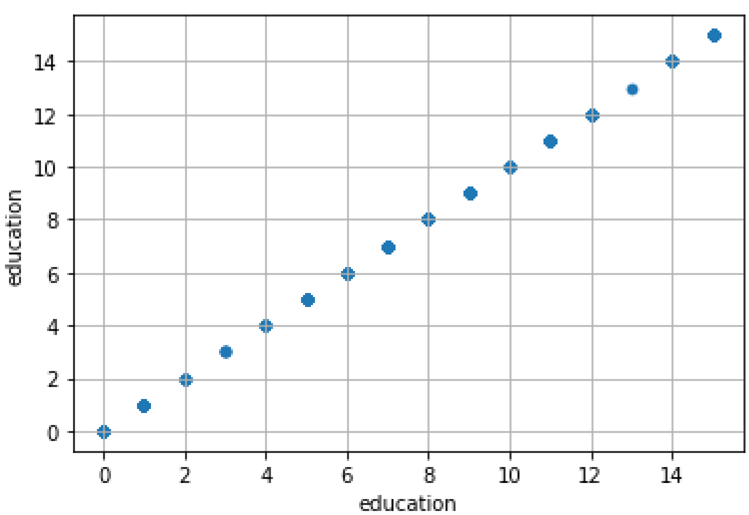


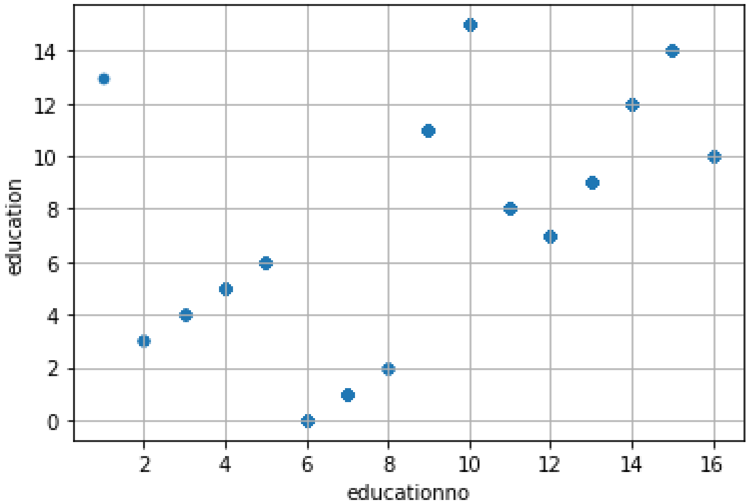
In [24]:

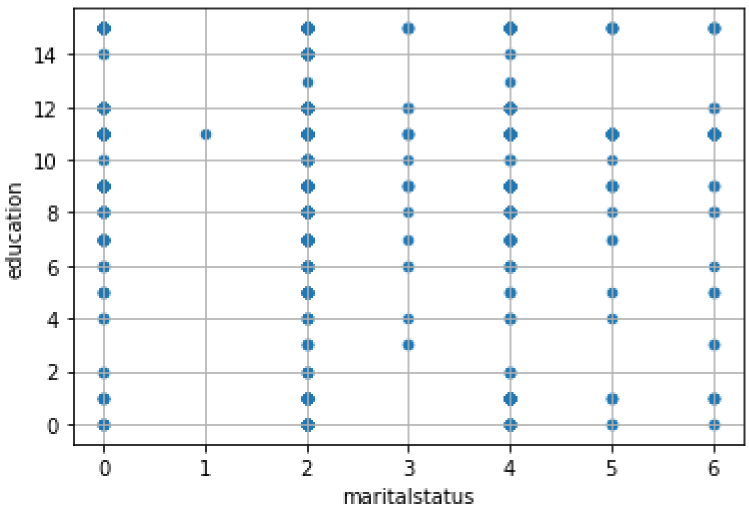
```
for i in train.describe().columns[:-2]:
    train.plot.scatter(i,'education',grid=True)
```

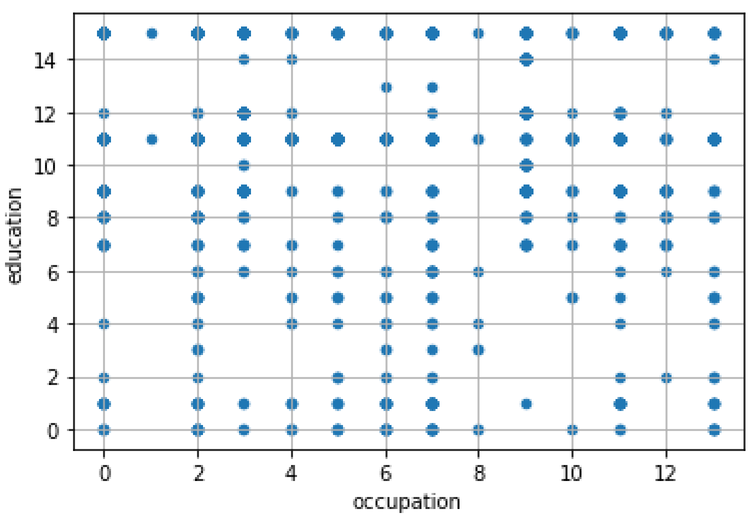

education
age

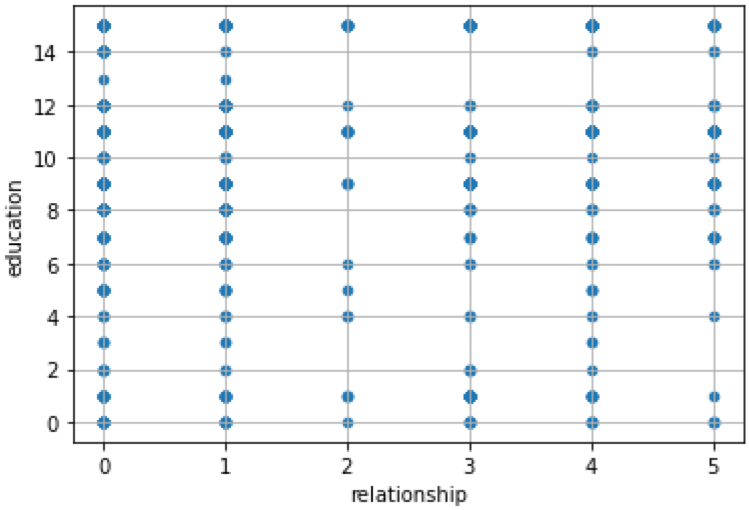

education
workclass

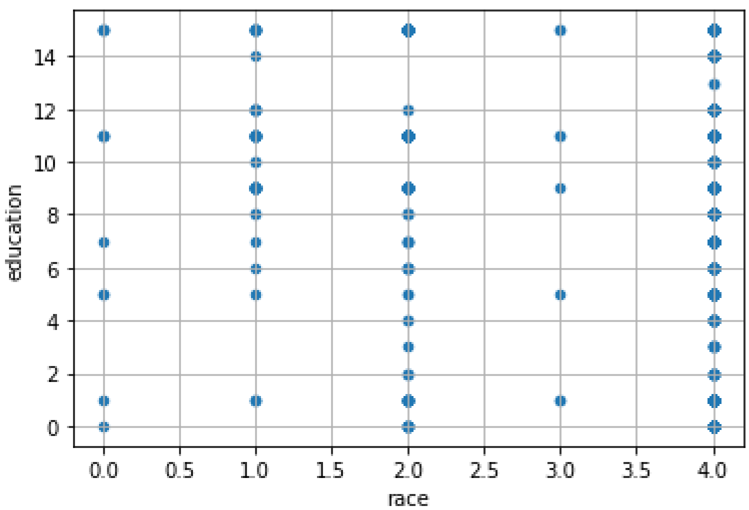

education
education

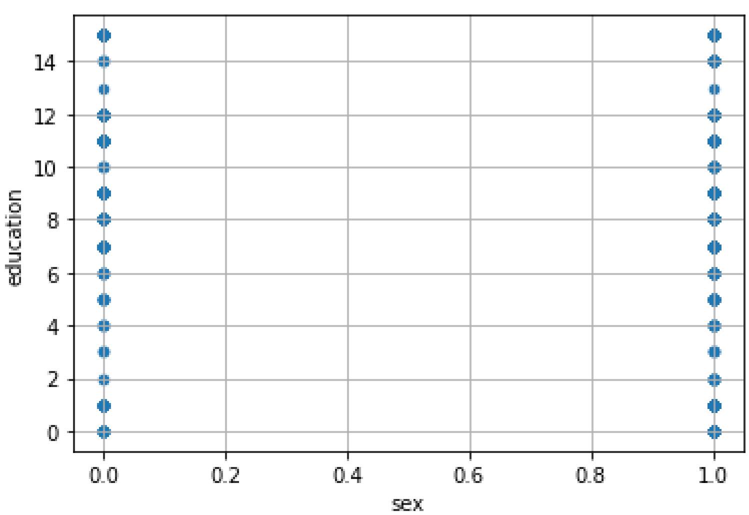

education
educationno
0
2
4
6
8
10
12
14
16

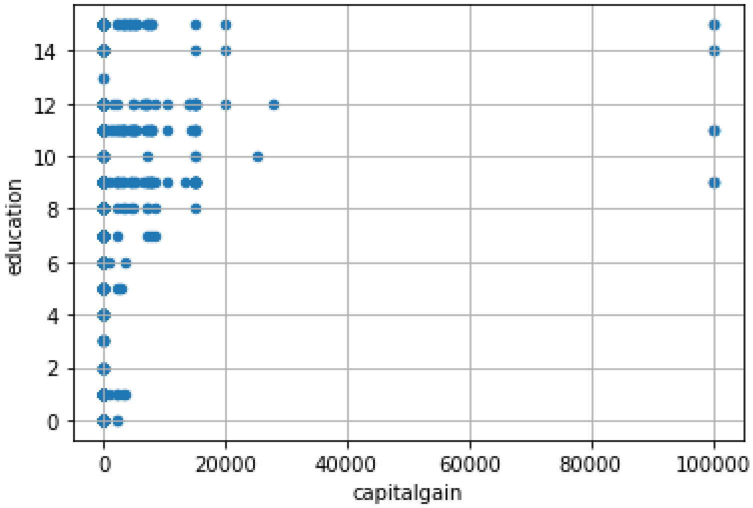

education
maritalstatus
0
1
2
3
4
5
6

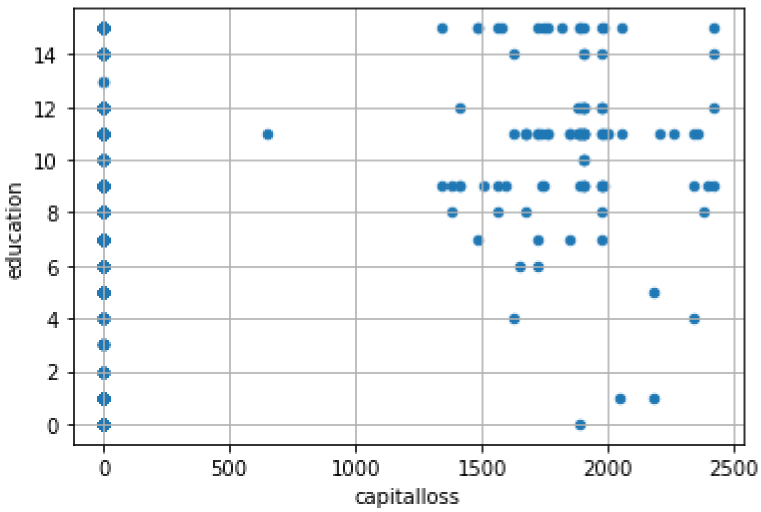

education
occupation
0
2
4
6
8
10
12

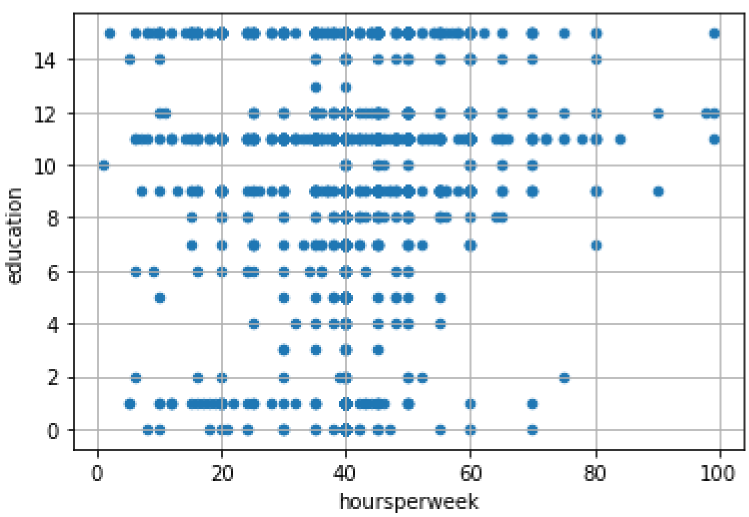

education
relationship


education
race


education
sex







In [25]:

```
train.corr()
```

Out[25]:

	age	workclass	education	educationno	maritalstatus	occupation	relationship	
age	1.000000	0.080136	-0.004007	0.014781	-0.249467	-0.004634	-0.216588	0
workclass	0.080136	1.000000	0.029167	0.068866	-0.043219	0.033209	-0.074186	0
education	-0.004007	0.029167	1.000000	0.328746	-0.047668	-0.028564	-0.033833	0

	age	workclass	education	educationno	maritalstatus	occupation	relationship	
educationno	0.014781	0.068866	0.328746	1.000000	-0.062303	0.098459	-0.091217	0
maritalstatus	-0.249467	-0.043219	-0.047668	-0.062303	1.000000	0.075036	0.157226	-0
occupation	-0.004634	0.033209	-0.028564	0.098459	0.075036	1.000000	-0.065478	0
relationship	-0.216588	-0.074186	-0.033833	-0.091217	0.157226	-0.065478	1.000000	-0
race	0.015168	0.074418	0.031098	0.075867	-0.083280	0.035830	-0.100663	1
sex	0.050730	0.087332	-0.004879	0.034123	-0.078456	0.072483	-0.557999	0
capitalgain	0.081112	0.038314	0.035363	0.095804	-0.044395	0.016453	-0.052849	0
capitalloss	0.058997	-0.003069	0.012082	0.062601	-0.016550	-0.015165	-0.042243	0
hoursperweek	0.114429	0.031221	0.060470	0.172302	-0.187437	0.032509	-0.256052	0
native	-0.001914	-0.036263	0.085718	0.057602	0.002006	-0.000106	-0.054397	0
Salary	0.231176	0.064561	0.051282	0.308324	-0.199289	0.026793	-0.211663	0

In [26]:

```
test.corr()
```

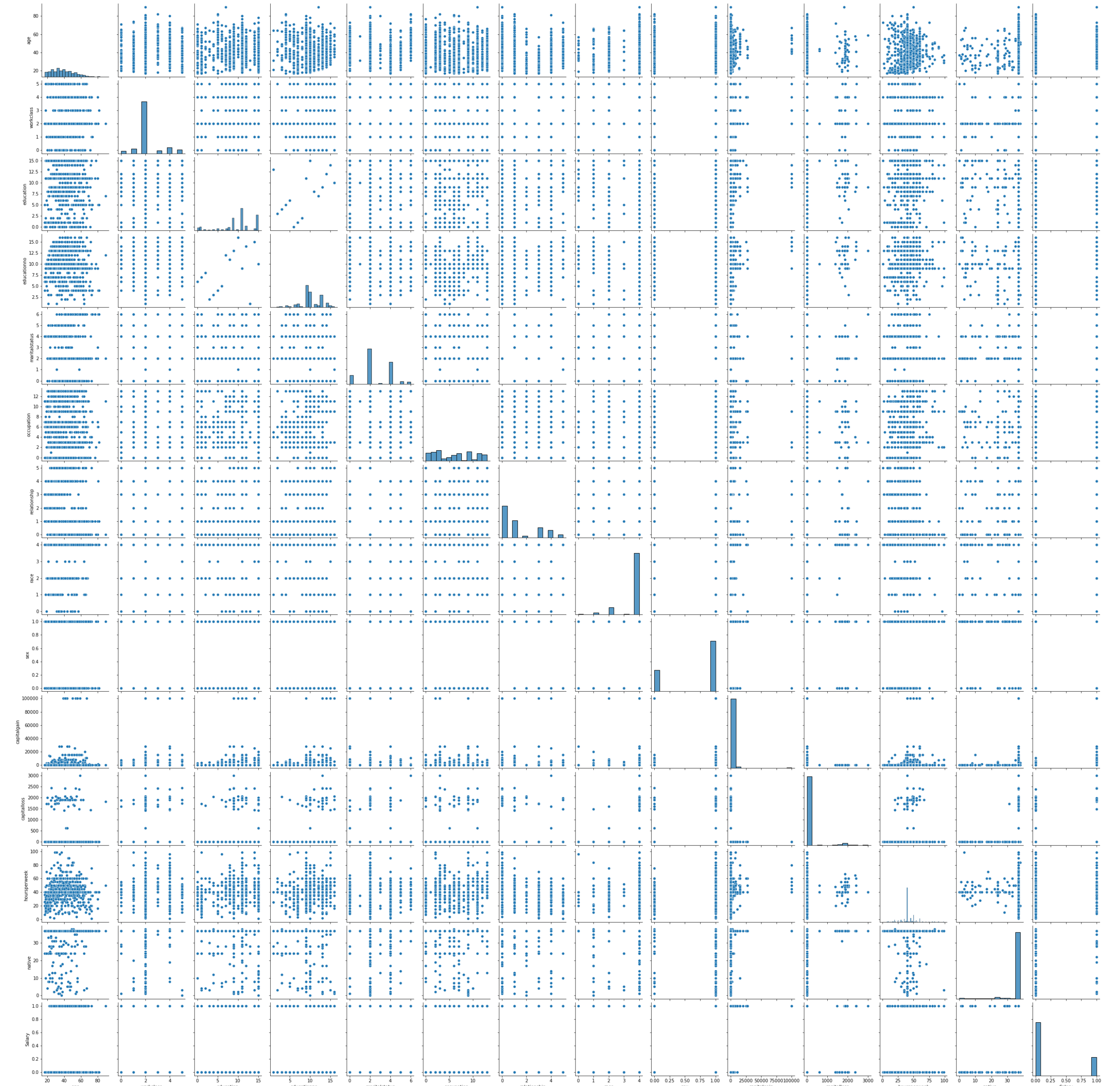
Out[26]:

	age	workclass	education	educationno	maritalstatus	occupation	relationship	
age	1.000000	0.132698	-0.058456	-0.028238	-0.299098	-0.035747	-0.253358	0
workclass	0.132698	1.000000	0.068824	0.066220	-0.060726	0.011914	-0.082095	0
education	-0.058456	0.068824	1.000000	0.409201	-0.039852	-0.023030	-0.024427	-0
educationno	-0.028238	0.066220	0.409201	1.000000	-0.091147	0.077972	-0.050679	0
maritalstatus	-0.299098	-0.060726	-0.039852	-0.091147	1.000000	-0.012545	0.197796	-0
occupation	-0.035747	0.011914	-0.023030	0.077972	-0.012545	1.000000	-0.063657	0
relationship	-0.253358	-0.082095	-0.024427	-0.050679	0.197796	-0.063657	1.000000	-0
race	0.003103	0.065081	-0.005501	0.072135	-0.085572	0.037160	-0.158517	1
sex	0.074865	0.049687	-0.011597	-0.011529	-0.147099	0.073262	-0.588370	0
capitalgain	0.107361	0.053677	0.060160	0.170971	-0.071843	0.004247	-0.060524	0
capitalloss	0.057418	0.062026	0.038734	0.111713	-0.018926	-0.038230	-0.037262	0
hoursperweek	0.106105	0.077106	0.063598	0.145317	-0.161722	0.018785	-0.298815	0
native	0.018697	0.043338	0.069760	0.106144	0.005488	0.004193	0.015321	0
Salary	0.226220	0.077615	0.119016	0.313422	-0.226209	0.018910	-0.259728	0

In [27]:

```
sns.pairplot(test)
```

Out[27]: <seaborn.axisgrid.PairGrid at 0x1f0ee9dd520>



In [28]:
```python
X_train=train.iloc[:,:-1]
X_train
```

Out[28]:

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	sex	capit
0	39	5	9	13	4	0	1	4	1	
1	50	4	9	13	2	3	0	4	1	
2	38	2	11	9	0	5	1	4	1	
3	53	2	1	7	2	5	0	2	1	
4	28	2	9	13	2	9	5	2	0	
...	...	...	...	...	...	...	...	...	...	
1995	33	2	11	9	5	10	3	4	0	
1996	41	2	11	9	2	6	0	4	1	

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	sex	capit
1997	51	2	6	5	2	13	0	4	1	
1998	42	2	11	9	2	11	0	4	1	
1999	27	2	9	13	2	9	5	4	0	

2000 rows × 13 columns

In [29]:

```
y_train=train.iloc[:,-1]
y_train
```

Out[29]:

```
0       0
1       0
2       0
3       0
4       0
       ..
1995    0
1996    0
1997    0
1998    0
1999    1
Name: Salary, Length: 2000, dtype: int32
```

In [30]:

```
X_test=test.iloc[:,:-1]
X_test
```

Out[30]:

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	sex	capit
0	25	2	1	7	4	6	3	2	1	
1	38	2	11	9	2	4	0	4	1	
2	28	1	7	12	2	10	0	4	1	
3	44	2	15	10	2	6	0	2	1	
4	34	2	0	6	4	7	1	4	1	
...	...	...	...	...	...	...	...	...	...	
1295	66	4	15	10	2	13	0	2	1	
1296	40	2	15	10	4	2	2	0	1	
1297	37	2	4	3	2	6	0	4	1	
1298	34	2	9	13	2	11	0	4	1	
1299	41	4	12	14	6	9	1	4	0	

1300 rows × 13 columns

In [31]:

```
y_test = test.iloc[:,-1]
```

```
y_test
```

Out[31]:
```
0       0
1       0
2       1
3       1
4       0
       ..
1295    0
1296    0
1297    0
1298    0
1299    0
Name: Salary, Length: 1300, dtype: int32
```

In [32]:
```
X_train.shape, y_train.shape, X_test.shape, y_test.shape
```

Out[32]:
```
((2000, 13), (2000,), (1300, 13), (1300,))
```

In [33]:
```
model = SVC()

model.fit(X_train, y_train)
```

Out[33]:
```
SVC()
```

In [34]:
```
y_pred = model.predict(X_test)
y_pred
```

Out[34]:
```
array([0, 0, 0, ..., 0, 0, 0])
```

In [35]:
```
print(confusion_matrix(y_test, y_pred))
```

```
[[961   5]
 [267  67]]
```

In [36]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.78      0.99      0.88       966
           1       0.93      0.20      0.33       334

    accuracy                           0.79      1300
   macro avg       0.86      0.60      0.60      1300
weighted avg       0.82      0.79      0.74      1300
```

In [37]:
```
param_grid = {'C' : [1, 5, 10, 15, 20], 'gamma' : [1, 0.1, 0.01, 0.001, 0.0001], 'kerne
```

In [38]:
```
grid = GridSearchCV(SVC(), param_grid, refit = True, verbose = 3,  cv = 5)
```

In [39]:
```
grid.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 25 candidates, totalling 125 fits
[CV 1/5] END .........C=1, gamma=1, kernel=rbf;, score=0.752 total time=   0.5s
[CV 2/5] END .........C=1, gamma=1, kernel=rbf;, score=0.750 total time=   0.5s
[CV 3/5] END .........C=1, gamma=1, kernel=rbf;, score=0.745 total time=   0.4s
[CV 4/5] END .........C=1, gamma=1, kernel=rbf;, score=0.745 total time=   0.5s
[CV 5/5] END .........C=1, gamma=1, kernel=rbf;, score=0.743 total time=   0.4s
[CV 1/5] END .......C=1, gamma=0.1, kernel=rbf;, score=0.748 total time=   0.5s
[CV 2/5] END .......C=1, gamma=0.1, kernel=rbf;, score=0.738 total time=   0.4s
[CV 3/5] END .......C=1, gamma=0.1, kernel=rbf;, score=0.738 total time=   0.4s
[CV 4/5] END .......C=1, gamma=0.1, kernel=rbf;, score=0.755 total time=   0.4s
[CV 5/5] END .......C=1, gamma=0.1, kernel=rbf;, score=0.748 total time=   0.5s
[CV 1/5] END ......C=1, gamma=0.01, kernel=rbf;, score=0.775 total time=   0.3s
[CV 2/5] END ......C=1, gamma=0.01, kernel=rbf;, score=0.805 total time=   0.2s
[CV 3/5] END ......C=1, gamma=0.01, kernel=rbf;, score=0.812 total time=   0.2s
[CV 4/5] END ......C=1, gamma=0.01, kernel=rbf;, score=0.805 total time=   0.2s
[CV 5/5] END ......C=1, gamma=0.01, kernel=rbf;, score=0.802 total time=   0.2s
[CV 1/5] END .....C=1, gamma=0.001, kernel=rbf;, score=0.802 total time=   0.4s
[CV 2/5] END .....C=1, gamma=0.001, kernel=rbf;, score=0.812 total time=   0.2s
[CV 3/5] END .....C=1, gamma=0.001, kernel=rbf;, score=0.815 total time=   0.1s
[CV 4/5] END .....C=1, gamma=0.001, kernel=rbf;, score=0.815 total time=   0.2s
[CV 5/5] END .....C=1, gamma=0.001, kernel=rbf;, score=0.818 total time=   0.1s
[CV 1/5] END ....C=1, gamma=0.0001, kernel=rbf;, score=0.802 total time=   0.1s
[CV 2/5] END ....C=1, gamma=0.0001, kernel=rbf;, score=0.807 total time=   0.1s
[CV 3/5] END ....C=1, gamma=0.0001, kernel=rbf;, score=0.815 total time=   0.1s
[CV 4/5] END ....C=1, gamma=0.0001, kernel=rbf;, score=0.815 total time=   0.5s
[CV 5/5] END ....C=1, gamma=0.0001, kernel=rbf;, score=0.815 total time=   0.1s
[CV 1/5] END .........C=5, gamma=1, kernel=rbf;, score=0.748 total time=   0.5s
[CV 2/5] END .........C=5, gamma=1, kernel=rbf;, score=0.745 total time=   0.6s
[CV 3/5] END .........C=5, gamma=1, kernel=rbf;, score=0.743 total time=   0.5s
[CV 4/5] END .........C=5, gamma=1, kernel=rbf;, score=0.750 total time=   0.5s
[CV 5/5] END .........C=5, gamma=1, kernel=rbf;, score=0.750 total time=   0.5s
[CV 1/5] END .......C=5, gamma=0.1, kernel=rbf;, score=0.728 total time=   0.6s
[CV 2/5] END .......C=5, gamma=0.1, kernel=rbf;, score=0.730 total time=   0.4s
[CV 3/5] END .......C=5, gamma=0.1, kernel=rbf;, score=0.730 total time=   0.4s
[CV 4/5] END .......C=5, gamma=0.1, kernel=rbf;, score=0.738 total time=   0.5s
[CV 5/5] END .......C=5, gamma=0.1, kernel=rbf;, score=0.733 total time=   0.4s
[CV 1/5] END ......C=5, gamma=0.01, kernel=rbf;, score=0.782 total time=   0.2s
[CV 2/5] END ......C=5, gamma=0.01, kernel=rbf;, score=0.807 total time=   0.3s
[CV 3/5] END ......C=5, gamma=0.01, kernel=rbf;, score=0.815 total time=   0.3s
[CV 4/5] END ......C=5, gamma=0.01, kernel=rbf;, score=0.810 total time=   0.2s
[CV 5/5] END ......C=5, gamma=0.01, kernel=rbf;, score=0.800 total time=   0.4s
[CV 1/5] END .....C=5, gamma=0.001, kernel=rbf;, score=0.802 total time=   0.2s
[CV 2/5] END .....C=5, gamma=0.001, kernel=rbf;, score=0.815 total time=   0.2s
[CV 3/5] END .....C=5, gamma=0.001, kernel=rbf;, score=0.815 total time=   0.2s
[CV 4/5] END .....C=5, gamma=0.001, kernel=rbf;, score=0.815 total time=   0.3s
[CV 5/5] END .....C=5, gamma=0.001, kernel=rbf;, score=0.815 total time=   0.2s
[CV 1/5] END ....C=5, gamma=0.0001, kernel=rbf;, score=0.800 total time=   0.5s
[CV 2/5] END ....C=5, gamma=0.0001, kernel=rbf;, score=0.807 total time=   0.3s
[CV 3/5] END ....C=5, gamma=0.0001, kernel=rbf;, score=0.812 total time=   0.2s
[CV 4/5] END ....C=5, gamma=0.0001, kernel=rbf;, score=0.818 total time=   0.2s
[CV 5/5] END ....C=5, gamma=0.0001, kernel=rbf;, score=0.812 total time=   0.4s
[CV 1/5] END ........C=10, gamma=1, kernel=rbf;, score=0.748 total time=   0.8s
[CV 2/5] END ........C=10, gamma=1, kernel=rbf;, score=0.745 total time=   0.6s
[CV 3/5] END ........C=10, gamma=1, kernel=rbf;, score=0.743 total time=   0.8s
[CV 4/5] END ........C=10, gamma=1, kernel=rbf;, score=0.750 total time=   0.6s
[CV 5/5] END ........C=10, gamma=1, kernel=rbf;, score=0.750 total time=   0.8s
[CV 1/5] END ......C=10, gamma=0.1, kernel=rbf;, score=0.728 total time=   0.4s
[CV 2/5] END ......C=10, gamma=0.1, kernel=rbf;, score=0.733 total time=   0.6s
```

```
[CV 3/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.728 total time=   0.4s
[CV 4/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.745 total time=   0.4s
[CV 5/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.733 total time=   0.5s
[CV 1/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.787 total time=   0.3s
[CV 2/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.815 total time=   0.3s
[CV 3/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.792 total time=   0.2s
[CV 4/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.812 total time=   0.3s
[CV 5/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.790 total time=   0.2s
[CV 1/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.818 total time=   0.1s
[CV 2/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.815 total time=   0.2s
[CV 3/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.823 total time=   0.2s
[CV 4/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.812 total time=   0.3s
[CV 5/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.825 total time=   0.2s
[CV 1/5] END ....C=10, gamma=0.0001, kernel=rbf;, score=0.800 total time=   0.2s
[CV 2/5] END ....C=10, gamma=0.0001, kernel=rbf;, score=0.807 total time=   0.2s
[CV 3/5] END ....C=10, gamma=0.0001, kernel=rbf;, score=0.815 total time=   0.2s
[CV 4/5] END ....C=10, gamma=0.0001, kernel=rbf;, score=0.818 total time=   0.2s
[CV 5/5] END ....C=10, gamma=0.0001, kernel=rbf;, score=0.812 total time=   0.2s
[CV 1/5] END .........C=15, gamma=1, kernel=rbf;, score=0.748 total time=   0.5s
[CV 2/5] END .........C=15, gamma=1, kernel=rbf;, score=0.745 total time=   0.5s
[CV 3/5] END .........C=15, gamma=1, kernel=rbf;, score=0.743 total time=   0.5s
[CV 4/5] END .........C=15, gamma=1, kernel=rbf;, score=0.750 total time=   0.5s
[CV 5/5] END .........C=15, gamma=1, kernel=rbf;, score=0.750 total time=   0.5s
[CV 1/5] END .......C=15, gamma=0.1, kernel=rbf;, score=0.720 total time=   0.5s
[CV 2/5] END .......C=15, gamma=0.1, kernel=rbf;, score=0.730 total time=   0.5s
[CV 3/5] END .......C=15, gamma=0.1, kernel=rbf;, score=0.725 total time=   0.4s
[CV 4/5] END .......C=15, gamma=0.1, kernel=rbf;, score=0.745 total time=   0.5s
[CV 5/5] END .......C=15, gamma=0.1, kernel=rbf;, score=0.735 total time=   0.4s
[CV 1/5] END ......C=15, gamma=0.01, kernel=rbf;, score=0.777 total time=   0.2s
[CV 2/5] END ......C=15, gamma=0.01, kernel=rbf;, score=0.818 total time=   0.2s
[CV 3/5] END ......C=15, gamma=0.01, kernel=rbf;, score=0.792 total time=   0.2s
[CV 4/5] END ......C=15, gamma=0.01, kernel=rbf;, score=0.800 total time=   0.2s
[CV 5/5] END ......C=15, gamma=0.01, kernel=rbf;, score=0.792 total time=   0.2s
[CV 1/5] END .....C=15, gamma=0.001, kernel=rbf;, score=0.818 total time=   0.2s
[CV 2/5] END .....C=15, gamma=0.001, kernel=rbf;, score=0.820 total time=   0.2s
[CV 3/5] END .....C=15, gamma=0.001, kernel=rbf;, score=0.825 total time=   0.2s
[CV 4/5] END .....C=15, gamma=0.001, kernel=rbf;, score=0.812 total time=   0.1s
[CV 5/5] END .....C=15, gamma=0.001, kernel=rbf;, score=0.825 total time=   0.1s
[CV 1/5] END ....C=15, gamma=0.0001, kernel=rbf;, score=0.800 total time=   0.2s
[CV 2/5] END ....C=15, gamma=0.0001, kernel=rbf;, score=0.807 total time=   0.2s
[CV 3/5] END ....C=15, gamma=0.0001, kernel=rbf;, score=0.815 total time=   0.2s
[CV 4/5] END ....C=15, gamma=0.0001, kernel=rbf;, score=0.815 total time=   0.1s
[CV 5/5] END ....C=15, gamma=0.0001, kernel=rbf;, score=0.812 total time=   0.2s
[CV 1/5] END .........C=20, gamma=1, kernel=rbf;, score=0.748 total time=   0.5s
[CV 2/5] END .........C=20, gamma=1, kernel=rbf;, score=0.745 total time=   0.5s
[CV 3/5] END .........C=20, gamma=1, kernel=rbf;, score=0.743 total time=   0.5s
[CV 4/5] END .........C=20, gamma=1, kernel=rbf;, score=0.750 total time=   0.5s
[CV 5/5] END .........C=20, gamma=1, kernel=rbf;, score=0.750 total time=   0.6s
[CV 1/5] END .......C=20, gamma=0.1, kernel=rbf;, score=0.723 total time=   0.5s
[CV 2/5] END .......C=20, gamma=0.1, kernel=rbf;, score=0.730 total time=   0.4s
[CV 3/5] END .......C=20, gamma=0.1, kernel=rbf;, score=0.725 total time=   0.4s
[CV 4/5] END .......C=20, gamma=0.1, kernel=rbf;, score=0.745 total time=   0.4s
[CV 5/5] END .......C=20, gamma=0.1, kernel=rbf;, score=0.738 total time=   0.5s
[CV 1/5] END ......C=20, gamma=0.01, kernel=rbf;, score=0.767 total time=   0.2s
[CV 2/5] END ......C=20, gamma=0.01, kernel=rbf;, score=0.807 total time=   0.2s
[CV 3/5] END ......C=20, gamma=0.01, kernel=rbf;, score=0.787 total time=   0.2s
[CV 4/5] END ......C=20, gamma=0.01, kernel=rbf;, score=0.802 total time=   0.3s
[CV 5/5] END ......C=20, gamma=0.01, kernel=rbf;, score=0.785 total time=   0.2s
[CV 1/5] END .....C=20, gamma=0.001, kernel=rbf;, score=0.815 total time=   0.2s
[CV 2/5] END .....C=20, gamma=0.001, kernel=rbf;, score=0.818 total time=   0.2s
```

```
[CV 3/5] END .....C=20, gamma=0.001, kernel=rbf;, score=0.825 total time=   0.2s
[CV 4/5] END .....C=20, gamma=0.001, kernel=rbf;, score=0.810 total time=   0.2s
[CV 5/5] END .....C=20, gamma=0.001, kernel=rbf;, score=0.823 total time=   0.2s
[CV 1/5] END ....C=20, gamma=0.0001, kernel=rbf;, score=0.800 total time=   0.2s
[CV 2/5] END ....C=20, gamma=0.0001, kernel=rbf;, score=0.807 total time=   0.2s
[CV 3/5] END ....C=20, gamma=0.0001, kernel=rbf;, score=0.815 total time=   0.1s
[CV 4/5] END ....C=20, gamma=0.0001, kernel=rbf;, score=0.818 total time=   0.2s
[CV 5/5] END ....C=20, gamma=0.0001, kernel=rbf;, score=0.815 total time=   0.1s
```

Out[39]:
```
GridSearchCV(cv=5, estimator=SVC(),
             param_grid={'C': [1, 5, 10, 15, 20],
                         'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
                         'kernel': ['rbf']},
             verbose=3)
```

In [40]:
```
grid.best_params_
```

Out[40]:
```
{'C': 15, 'gamma': 0.001, 'kernel': 'rbf'}
```

In [41]:
```
grid_pred = grid.predict(X_test)
```

In [42]:
```
grid_pred
```

Out[42]:
```
array([0, 0, 0, ..., 0, 0, 0])
```

In [43]:
```
print(confusion_matrix(y_test, grid_pred))
```

```
[[938  28]
 [198 136]]
```

In [44]:
```
print(classification_report(y_test, grid_pred))
```

```
              precision    recall  f1-score   support

           0       0.83      0.97      0.89       966
           1       0.83      0.41      0.55       334

    accuracy                           0.83      1300
   macro avg       0.83      0.69      0.72      1300
weighted avg       0.83      0.83      0.80      1300
```

In [ ]: