In [49]:
```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification
from sklearn.model_selection import GridSearchCV
import pandas as pd
import numpy as np
from sklearn.metrics import confusion_matrix
```

In [24]:
```python
datas = pd.read_csv('/Users/uliaandreeva/Desktop/heart.csv')
datas.head(10)
```

Out[24]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1

In [14]:
```python
np.random.seed(0)

rf = RandomForestClassifier(10, max_depth=5)
```

In [15]:
```python
y_train = datas['target']
X_train = datas.drop(['target'], axis=1)
```

In [17]:
```python
params = {
    "n_estimators": [10, 20, 30],
    'max_depth' : range(3, 8),
    'min_samples_split': [2, 6, 10],
    'min_samples_leaf' : [1, 5, 10],}
```

In [18]:
```python
search = GridSearchCV(rf, params, cv=5)
search.fit(X_train, y_train)
```

Out[18]:
```
GridSearchCV(cv=5,
             estimator=RandomForestClassifier(max_depth=5, n_estimators=10),
             param_grid={'max_depth': range(3, 8),
                         'min_samples_leaf': [1, 5, 10],
                         'min_samples_split': [2, 6, 10],
                         'n_estimators': [10, 20, 30]})
```

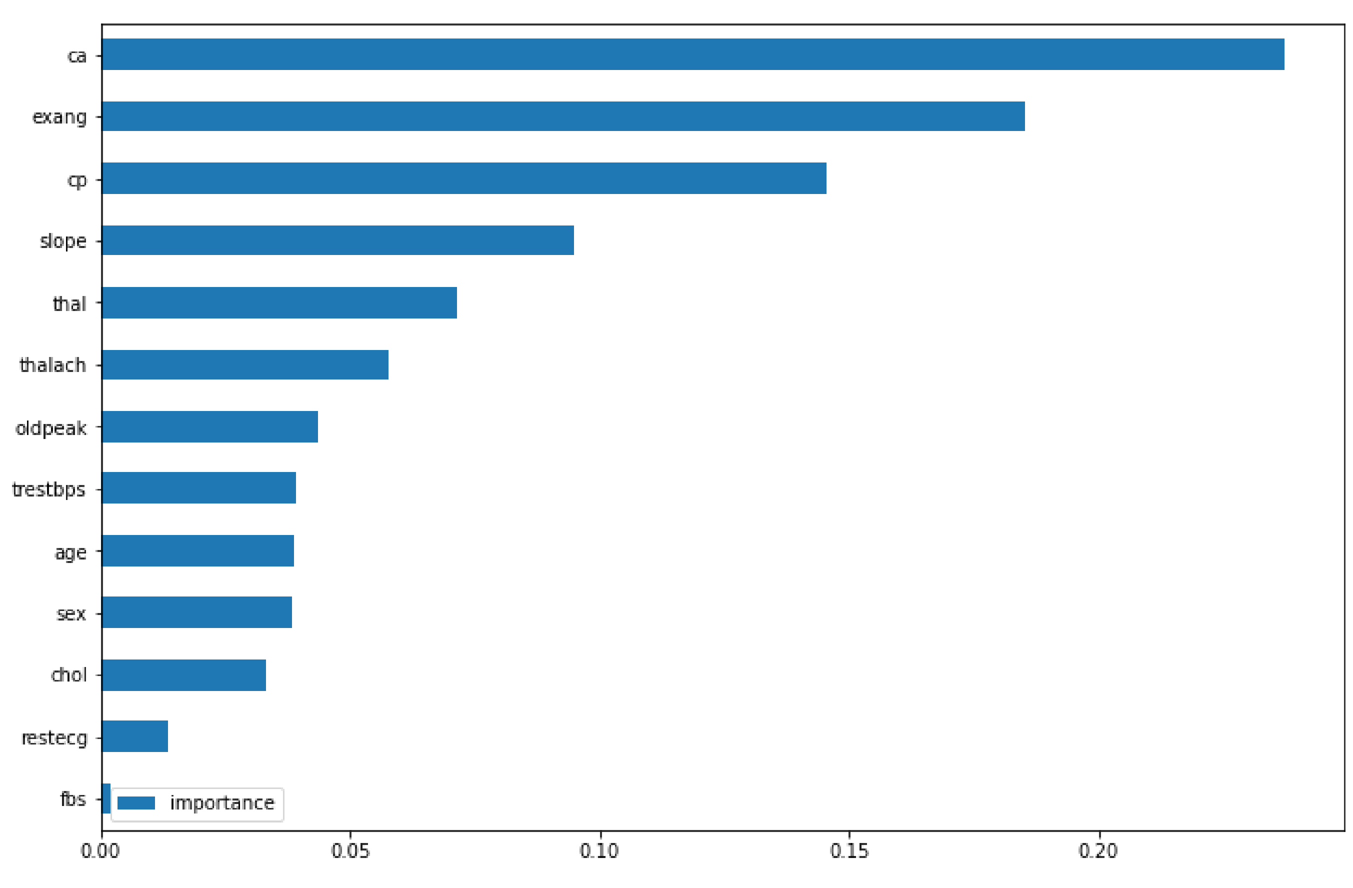
In [19]:
```python
rf = search.best_estimator_
```

In [22]:
```python
imp = pd.DataFrame(rf.feature_importances_, index=X_train.columns, columns=['importanc
print(imp.sort_values('importance', ascending=False).head(5))
```

```
       importance
ca       0.237248
exang    0.185174
cp       0.145520
slope    0.094971
thal     0.071552
```

In [23]:
```python
imp.sort_values('importance').plot(kind='barh', figsize=(12, 8))
```

Out[23]: <AxesSubplot:>



In [25]:
```python
gribs = pd.read_csv('/Users/uliaandreeva/Desktop/training_mush.csv')
gribs.head()
```

Out[25]:

	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	...	stalk-color-above-ring	stalk-color-below-ring
0	2	0	3	1	5	1	0	0	9	1	...	3	7
1	2	0	4	0	5	1	0	1	10	0	...	7	4
2	2	0	3	0	2	1	0	0	7	0	...	0	4
3	0	0	3	0	5	1	1	0	2	0	...	7	7
4	2	3	3	1	5	1	0	0	10	1	...	3	6

5 rows × 23 columns

In [29]:
```python
rrf = RandomForestClassifier(random_state=0)
params = {
    "n_estimators": range(10, 51, 10),
    'max_depth' : range(1, 13, 2),
    'min_samples_split': range(2, 9, 2),
    'min_samples_leaf' : range(1, 8),}
```

In [38]:
```python
x_train = gribs.drop(['class'], axis=1)
y_train = gribs['class']
search = GridSearchCV(rrf, params, cv=3,  n_jobs=-1)
search.fit(x_train, y_train)
search.best_params_
```

Out[38]:
```
{'max_depth': 9,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'n_estimators': 10}
```

In [42]:
```python
rf = search.best_estimator_
imp = pd.DataFrame(rf.feature_importances_, index=x_train.columns, columns=['importanc
print(imp.sort_values('importance', ascending=False).head(25))
```

```
                          importance
odor                        0.188376
gill-color                  0.103861
stalk-root                  0.103793
spore-print-color           0.083564
gill-spacing                0.080840
ring-type                   0.070726
bruises                     0.070109
gill-size                   0.068461
stalk-surface-below-ring    0.048296
population                  0.043783
stalk-surface-above-ring    0.031802
habitat                     0.022611
cap-color                   0.021773
stalk-color-above-ring      0.017689
ring-number                 0.013910
veil-color                  0.010881
stalk-shape                 0.007563
stalk-color-below-ring      0.006253
cap-surface                 0.003130
cap-shape                   0.002577
veil-type                   0.000000
gill-attachment             0.000000
```

In [45]:
```python
test_data = pd.read_csv('/Users/uliaandreeva/Desktop/testing_mush.csv')
test_data.head()
```

Out[45]:

	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	...	stalk-surface-below-ring	stalk-color-above-ring
0	0	3	8	1	3	1	0	0	4	0	...	2	7
1	5	3	4	1	5	1	0	0	10	1	...	2	6
2	3	3	4	0	8	1	0	1	0	1	...	2	7
3	3	2	2	0	7	1	0	1	0	1	...	1	6
4	3	2	2	0	8	1	0	1	0	1	...	1	6

5 rows × 22 columns

In [47]:
```python
predictions = search.predict(test_data)
predictions
```

Out[47]: `array([0, 0, 1, ..., 1, 0, 1])`

In [48]:
```python
answer = 0
for i in predictions:
    if i == 1:
        answer += 1
    else:
        answer = answer
answer
```

Out[48]: 976

In [51]:
```python
right_anw = pd.read_csv('/Users/uliaandreeva/Desktop/testing_y_mush.csv')
right_anw.head(10)
```

Out[51]:

	class
0	0
1	0
2	1
3	1
4	1
5	0
6	1
7	0
8	1
9	1

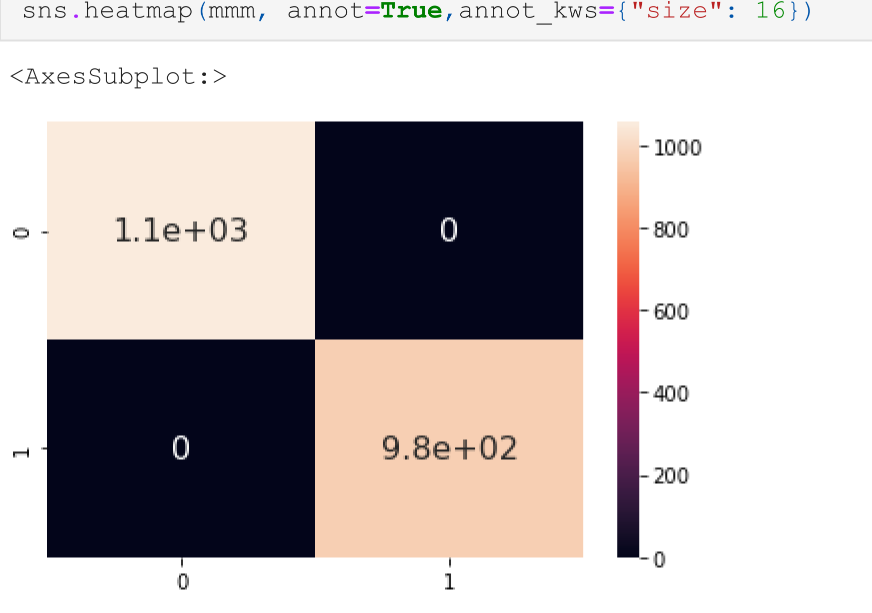
In [53]:
```python
mmm = confusion_matrix(right_anw, predictions)
mmm
```

Out[53]:
```
array([[1055,    0],
       [   0,  976]])
```

In [54]:
```python
import seaborn as sns

sns.heatmap(mmm, annot=True,annot_kws={"size": 16})
```

Out[54]: <AxesSubplot:>



In [55]:
```python
ships = pd.read_csv('/Users/uliaandreeva/Desktop/invasion.csv')
ships.head()
```

Out[55]:

	class	g_reflection	i_reflection	speed	brightness	time_of_observance	volume
0	transport	2.190672	6.716633	62.168208	0.347465	158221	44.932446
1	transport	3.453276	8.995909	62.994707	0.590094	385972	41.568300
2	transport	2.432994	6.938691	62.245807	0.329288	446482	40.123467
3	fighter	6.083763	3.019459	18.474555	0.174738	210125	11.384865
4	fighter	12.876769	2.452950	195.805771	0.150446	23109	11.328806

In [61]:
```python
ML_ships = RandomForestClassifier(random_state=0)
search = GridSearchCV(ML_ships, params, cv=3)
x_train = ships.drop(['class'], axis=1)
search.fit(x_train, ships['class'])
search.best_params_
```

Out[61]:
```
{'max_depth': 3,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'n_estimators': 10}
```

In [62]:
```python
test_ships = pd.read_csv('/Users/uliaandreeva/Desktop/operative_information.csv')
test_ships.head()
```

Out[62]:

	g_reflection	i_reflection	speed	brightness	time_of_observance	volume
0	7.516543	3.916691	513.954279	0.177247	105908	13.267224
1	4.322988	6.967689	63.752970	0.545922	277855	39.833130
2	4.595724	9.098297	62.233948	0.389201	160662	42.014556
3	2.689675	7.964869	62.475495	0.541081	162092	42.056829
4	8.075576	5.169719	336.441261	0.174757	466853	11.779813

In [63]:
```python
rf = search.best_estimator_
y_pred = rf.predict(test_ships)
y_pred
```

Out[63]:
```
array(['fighter', 'transport', 'transport', ..., 'transport', 'fighter',
       'transport'], dtype=object)
```

In [64]:
```python
s = pd.Series(y_pred)
s.groupby(s).count()
```

Out[64]:
```
cruiser      230
fighter      675
transport    595
dtype: int64
```

In [65]:
```python
imp = pd.DataFrame(rf.feature_importances_, index=x_train.columns, columns=['importanc
print(imp.sort_values('importance', ascending=False).head(25))
```

```
                    importance
brightness            0.363803
volume                0.257661
speed                 0.171441
i_reflection          0.124372
g_reflection          0.082723
time_of_observance    0.000000
```

In [ ]:

In [ ]:

In [67]:
```python
space = pd.read_csv('/Users/uliaandreeva/Desktop/space_can_be_a_dangerous_place.csv')
space.head()
```

Out[67]:

	r	phi	peradventure_index	dustiness	black_hole_is_near	buggers_were_noticed	nearby_syst
0	169.1	138.0	22.3212	0.706285	0	1	
1	11.1	148.0	1.4652	-0.410512	1	1	
2	274.6	201.0	36.2472	0.756457	1	1	
3	172.8	173.0	22.8096	0.035221	1	1	
4	223.3	222.0	29.4756	0.197271	0	1	

In [68]:
```python
sns.heatmap(space.corr(), annot=True)
```

Out[68]: <AxesSubplot:>

In [ ]:
```python
df.applymap()
df.apply()
df.transform()
```