In [25]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
from sklearn.linear_model import LogisticRegression
from sklearn import neighbors
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
```

In [26]:

```
#reading dataset
data = pd.read_csv('encoders (123).csv')
```

In [27]:

```
data
```

Out[27]:

	id	date	scanner	min	max	err	pixels	minf	maxf	errf	created_at1	up
0	12	2017-12-17	K219	35435	35933	1.40	6	35681	35688	0.02	2017-12-29	
1	30	2017-12-18	H161	35155	36382	3.43	14	35731	35761	0.08	2017-12-29	
2	47	2017-12-18	K211	35305	36042	2.07	43	35692	35739	0.13	2017-12-19	
3	48	2017-12-18	K212	35216	36225	2.82	61	35686	35726	0.11	2017-12-19	
4	49	2017-12-18	K220	35196	36259	2.98	11	35709	35724	0.04	2017-12-19	
...	...	...	...	...	...	...	...	...	...	...	...	
1916	2745	2018-03-07	K168	35185	35958	2.17	48	35541	35565	0.07	2018-03-08	
1917	2747	2018-03-07	K114	35220	36071	2.39	17	35715	35735	0.06	2018-03-08	
1918	2748	2018-03-07	K171	35109	35912	2.26	50	35504	35531	0.08	2018-03-08	
1919	2751	2018-03-07	K220	35181	36229	2.94	9	35702	35715	0.04	2018-03-08	
1920	2752	2018-03-07	K221	35288	36266	2.73	3	35734	35737	0.01	2018-03-08	

1921 rows × 12 columns

In [28]:

```
data["err1"] = (0.12 * data["err"])*100
print(data['err1'])
```

```
0       16.80
1       41.16
2       24.84
3       33.84
4       35.76
        ...
1916    26.04
1917    28.68
1918    27.12
1919    35.28
1920    32.76
Name: err1, Length: 1921, dtype: float64
```

In [29]:

```
data["errf1"] = (0.005 * data["errf"])*100
print(data['errf1'])
```

```
0       0.010
1       0.040
2       0.065
3       0.055
4       0.020
        ...
1916    0.035
1917    0.030
1918    0.040
1919    0.020
1920    0.005
Name: errf1, Length: 1921, dtype: float64
```

In [30]:

```
data.loc[data['err1'] >= 12, 'result_err1'] = 'never fail'
data.loc[data['err1'] < 12,'result_err1' ] = 'routinely fail'
data
```

Out[30]:

	id	date	scanner	min	max	err	pixels	minf	maxf	errf	created_at1	up
0	12	2017-12-17	K219	35435	35933	1.40	6	35681	35688	0.02	2017-12-29	
1	30	2017-12-18	H161	35155	36382	3.43	14	35731	35761	0.08	2017-12-29	
2	47	2017-12-18	K211	35305	36042	2.07	43	35692	35739	0.13	2017-12-19	
3	48	2017-12-18	K212	35216	36225	2.82	61	35686	35726	0.11	2017-12-19	
4	49	2017-12-18	K220	35196	36259	2.98	11	35709	35724	0.04	2017-12-19	
...	...	...	...	...	...	...	...	...	...	...	...	
1916	2745	2018-03-07	K168	35185	35958	2.17	48	35541	35565	0.07	2018-03-08	
1917	2747	2018-03-07	K114	35220	36071	2.39	17	35715	35735	0.06	2018-03-08	
1918	2748	2018-03-07	K171	35109	35912	2.26	50	35504	35531	0.08	2018-03-08	
1919	2751	2018-03-07	K220	35181	36229	2.94	9	35702	35715	0.04	2018-03-08	
1920	2752	2018-03-07	K221	35288	36266	2.73	3	35734	35737	0.01	2018-03-08	

1921 rows × 15 columns

In [31]:

```
data['result_err1'].value_counts()
```

Out[31]:

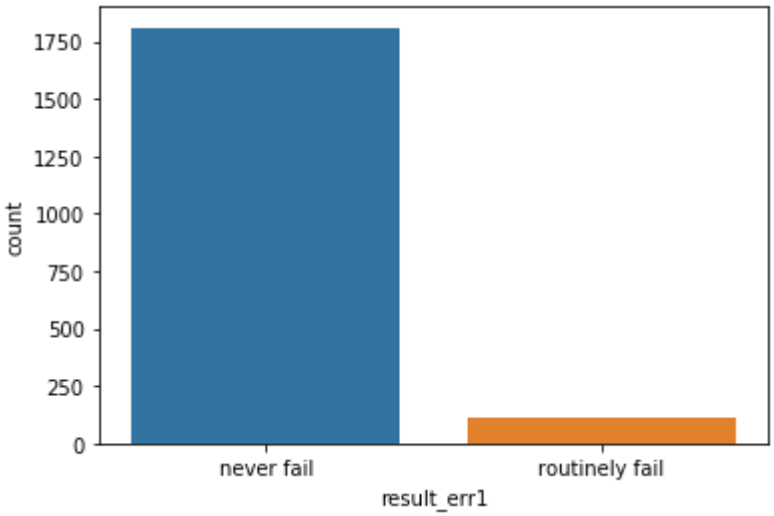
```
never fail        1810
routinely fail     111
Name: result_err1, dtype: int64
```

In [32]:

```
sns.countplot('result_err1', data=data)
```

Out[32]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x5611fc8>
```



In [33]:

```
data.loc[data['errf1'] >= 0.5, 'result_errf1'] = 'routinely fail'
data.loc[data['errf1'] < 0.5,'result_errf1' ] = 'never fail'
data
```

Out[33]:

	id	date	scanner	min	max	err	pixels	minf	maxf	errf	created_at1	up
0	12	2017-12-17	K219	35435	35933	1.40	6	35681	35688	0.02	2017-12-29	
1	30	2017-12-18	H161	35155	36382	3.43	14	35731	35761	0.08	2017-12-29	
2	47	2017-12-18	K211	35305	36042	2.07	43	35692	35739	0.13	2017-12-19	
3	48	2017-12-18	K212	35216	36225	2.82	61	35686	35726	0.11	2017-12-19	
4	49	2017-12-18	K220	35196	36259	2.98	11	35709	35724	0.04	2017-12-19	
...	...	...	...	...	...	...	...	...	...	...	...	
1916	2745	2018-03-07	K168	35185	35958	2.17	48	35541	35565	0.07	2018-03-08	
1917	2747	2018-03-07	K114	35220	36071	2.39	17	35715	35735	0.06	2018-03-08	
1918	2748	2018-03-07	K171	35109	35912	2.26	50	35504	35531	0.08	2018-03-08	
1919	2751	2018-03-07	K220	35181	36229	2.94	9	35702	35715	0.04	2018-03-08	
1920	2752	2018-03-07	K221	35288	36266	2.73	3	35734	35737	0.01	2018-03-08	

1921 rows × 16 columns

In [34]:

```
data['result_errf1'].value_counts()
```

Out[34]:

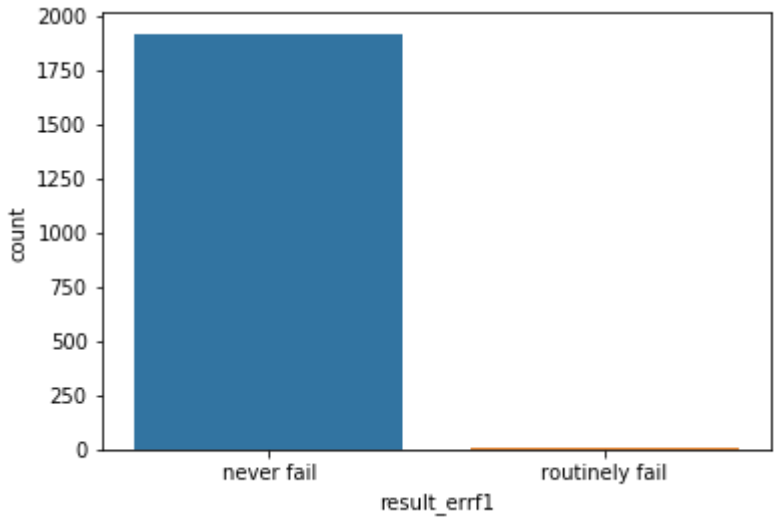
```
never fail        1916
routinely fail       5
Name: result_errf1, dtype: int64
```

In [35]:

```
sns.countplot('result_errf1', data=data)
```

Out[35]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x561a048>
```



In [36]:

```
data.columns
```

Out[36]:

```
Index(['id', 'date', 'scanner', 'min', 'max', 'err', 'pixels', 'minf', 'ma
xf',
       'errf', 'created_at1', 'updated_at2', 'err1', 'errf1', 'result_err
1',
       'result_errf1'],
      dtype='object')
```

In [37]:

```
#Sampling
cols = ['min', 'max','pixels']
X = data[cols]
print(X.head())
```

```
     min    max  pixels
0  35435  35933       6
1  35155  36382      14
2  35305  36042      43
3  35216  36225      61
4  35196  36259      11
```

In [38]:

```
y = data['result_err1']
print(y.head())
```

```
0    never fail
1    never fail
2    never fail
3    never fail
4    never fail
Name: result_err1, dtype: object
```

In [39]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y , test_size = 0.2)
```

In [40]:

```
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(1536, 3)
(1536,)
(385, 3)
(385,)
```

In [41]:

```
#Training
#model = LogisticRegression()
#model = neighbors.KNeighborsClassifier()
model = DecisionTreeClassifier(criterion='entropy',max_depth= 8)
#model = SVC(kernel='linear',  gamma = 10, C= 1)


model.fit(X_train,y_train)
#model = DecisionTreeClassifier(criterion='entropy',max_depth= 8)
#model.fit(X_train,y_train)
```

Out[41]:

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entrop
y',
                       max_depth=8, max_features=None, max_leaf_nodes=Non
e,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

In [42]:

```
#Testing
predicted = model.predict(X_test)
predicted
```

Out[42]:

```
array(['never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'routinely fail',
       'never fail', 'routinely fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'routinely fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'routinely fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'routinely fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'routinely fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'routinely fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'routinely fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'routinely fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
```

```
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'routinely fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'routinely fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail'], dtype=object)
```

In [43]:

```
#Evaluation
#Confusion Matrix
print(metrics.confusion_matrix(y_test, predicted))
```

```
[[367   0]
 [  0  18]]
```

In [44]:

```
#Classification Report
print(metrics.classification_report(y_test, predicted))
```

```
                precision    recall  f1-score   support

    never fail       1.00      1.00      1.00       367
routinely fail       1.00      1.00      1.00        18

      accuracy                           1.00       385
     macro avg       1.00      1.00      1.00       385
  weighted avg       1.00      1.00      1.00       385
```

In [45]:

```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = model, X = X_train,\
     y = y_train, cv = 10)
print("Accuracy Mean {} Accuracy Variance \
     {}".format(accuracies.mean(),accuracies.std()))
```

```
Accuracy Mean 0.9993506493506494 Accuracy Variance      0.0019480519480519
318
```

In [46]:

```
#Accuracy Score
from sklearn.metrics import accuracy_score

accuracy_score(y_test,predicted)
```

Out[46]:

```
1.0
```

In [47]:

```
from sklearn.model_selection import GridSearchCV
parameters = {'criterion':('gini', 'entropy'), 'max_depth':[8, 10,12]}
dt = DecisionTreeClassifier()
clf = GridSearchCV(dt, parameters)
clf.fit(X_train, y_train)
```

Out[47]:

```
GridSearchCV(cv=None, error_score=nan,
             estimator=DecisionTreeClassifier(ccp_alpha=0.0, class_weight=
None,
                                              criterion='gini', max_depth=
None,
                                              max_features=None,
                                              max_leaf_nodes=None,
                                              min_impurity_decrease=0.0,
                                              min_impurity_split=None,
                                              min_samples_leaf=1,
                                              min_samples_split=2,
                                              min_weight_fraction_leaf=0.
0,
                                              presort='deprecated',
                                              random_state=None,
                                              splitter='best'),
             iid='deprecated', n_jobs=None,
             param_grid={'criterion': ('gini', 'entropy'),
                         'max_depth': [8, 10, 12]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=Fals
e,
             scoring=None, verbose=0)
```

In [48]:

```
predicted = clf.predict(X_test)
predicted
```

Out[48]:

```
array(['never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'routinely fail',
       'never fail', 'routinely fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'routinely fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'routinely fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'routinely fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'routinely fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'routinely fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'routinely fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'routinely fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
```

```
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'routinely fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'routinely fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'routinely fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail', 'never fail', 'never fail', 'never fail',
       'never fail'], dtype=object)
```

In [49]:

```
from sklearn.metrics import accuracy_score

accuracy_score(y_test,predicted)
```

Out[49]:

```
1.0
```

In [ ]:

```

```

In [ ]:

```

```