In [1]:

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV, cross_val_score, StratifiedKFold, le
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

In [2]:

```python
data = pd.read_csv("file.csv")
```

In [3]:

```python
data.head()
```

Out[3]:

	0	1	2	3	4	5	6	7	
0	134217856	134217858	134217860	134217858	134217862	134217862	134217860	134217858	1342
1	134217856	134217860	167772300	134217858	134217862	167772296	134217860	134217860	1342
2	134217860	167772296	134217862	134217858	134217862	167772296	167772296	167772296	1677
3	134217862	167772302	167772296	167772296	167772296	167772294	134217860	167772298	1677
4	167772296	167772298	167772300	134217860	167772294	167772296	167772296	134217860	2348

5 rows × 2048 columns

In [4]:

```python
data.shape
```

Out[4]:

```
(2736, 2048)
```

In [5]:

```python
data.describe()
```

Out[5]:

	0	1	2	3	4	5	
count	2.736000e+03	2.736000e+03	2.736000e+03	2.736000e+03	2.736000e+03	2.736000e+03	2.73600
mean	1.662270e+08	1.623884e+08	1.565016e+08	1.448508e+08	1.456112e+08	1.451329e+08	1.45942
std	3.836914e+07	3.663680e+07	3.329157e+07	2.187373e+07	2.246150e+07	2.267064e+07	2.32928
min	1.342179e+08	1.342179e+08	1.342179e+08	1.342179e+08	1.342179e+08	1.342179e+08	1.34217
25%	1.342179e+08	1.342179e+08	1.342179e+08	1.342179e+08	1.342179e+08	1.342179e+08	1.34217
50%	1.677723e+08	1.342179e+08	1.342179e+08	1.342179e+08	1.342179e+08	1.342179e+08	1.34217
75%	1.677723e+08	1.677723e+08	1.677723e+08	1.342179e+08	1.677723e+08	1.342179e+08	1.67772
max	2.348812e+08	2.348812e+08	2.348812e+08	2.348812e+08	2.348812e+08	2.348812e+08	2.34881

8 rows × 2048 columns

In [6]:

```
X = data.iloc[:, :-1]
Y= data.iloc[:, -1]
```

In [7]:

```
validation_size = 0.20
seed = 7
num_folds = 10
scoring = 'accuracy'
X_train, X_validation, Y_train, Y_validation = train_test_split(X,Y, test_size=validat
```

In [8]:

```
num_trees = 100
max_features = 3
```

In [9]:

```
models = []
models.append(('LR', LogisticRegression()))
```

In [10]:

```
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('RF', RandomForestClassifier(n_estimators=num_trees, max_features=max_f
```

In [11]:

```
results = []
names = []
for name, model in models:
    kfold = KFold(n_splits=10)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy'
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

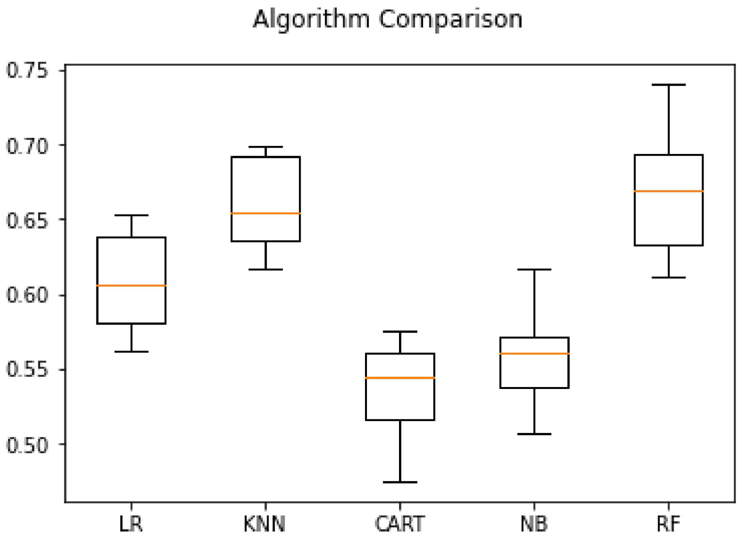
```
LR: 0.607400 (0.033069)
KNN: 0.659958 (0.029388)
CART: 0.535204 (0.033201)
NB: 0.558056 (0.030051)
RF: 0.667274 (0.039399)
```

In [12]:

```
import matplotlib.pyplot as plt
```

In [13]:

```
fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```



In [14]:

```
random_forest = RandomForestClassifier(n_estimators=250,max_features=5)
random_forest.fit(X_train, Y_train)
```

Out[14]:

```
RandomForestClassifier(max_features=5, n_estimators=250)
```

In [15]:

```
predictions = random_forest.predict(X_validation)
print("Accuracy: %s%%" % (100*accuracy_score(Y_validation, predictions)))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))

accuracy_score(Y_validation, predictions)
```

```
Accuracy: 64.78102189781022%
[[282   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   18]
 [ 17   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    2]
 [  9   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    3]
 [  6   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    2]
 [  7   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    2]
 [ 11   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    1]
 [  7   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    5]
 [  5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    1]
 [  3   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
    1]
 [ 19   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    5]
 [  3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    4]
 [  8   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    3]
 [  5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    3]
 [  2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    4]
 [  1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    7]
 [  2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    5]
 [  1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    4]
 [  1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    3]
 [ 11   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   73]]
              precision    recall  f1-score   support

   134217728       0.70      0.94      0.81       300
   134217730       0.00      0.00      0.00        19
   134217732       0.00      0.00      0.00        12
   134217734       0.00      0.00      0.00         9
   167772166       0.00      0.00      0.00         9
   167772168       0.00      0.00      0.00        12
   167772170       0.00      0.00      0.00        12
   167772172       0.00      0.00      0.00         6
   167772174       0.00      0.00      0.00         5
   167772190       0.00      0.00      0.00        24
   234881038       0.00      0.00      0.00         7
   234881040       0.00      0.00      0.00        11
   234881042       0.00      0.00      0.00         8
   234881044       0.00      0.00      0.00         6
   234881046       0.00      0.00      0.00         8
   234881048       0.00      0.00      0.00         7
   234881050       0.00      0.00      0.00         5
   234881052       0.00      0.00      0.00         4
   234881054       0.50      0.87      0.63        84
```

```
    accuracy                           0.65       548
   macro avg       0.06      0.10      0.08       548
weighted avg       0.46      0.65      0.54       548
```

```
C:\DIFF\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: Undefine
dMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\DIFF\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: Undefine
dMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\DIFF\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: Undefine
dMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

Out[15]:

```
0.6478102189781022
```

In [ ]: