In [2]:
```
import pandas as pd
import numpy as np

df = pd.read_csv('/Users/juanrquilesjr/Downloads/Kaggle_DataSets/glass.csv')
df.head()
```

Out[2]:

	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	Type
0	1.52101	13.64	4.49	1.10	71.78	0.06	8.75	0.0	0.0	1
1	1.51761	13.89	3.60	1.36	72.73	0.48	7.83	0.0	0.0	1
2	1.51618	13.53	3.55	1.54	72.99	0.39	7.78	0.0	0.0	1
3	1.51766	13.21	3.69	1.29	72.61	0.57	8.22	0.0	0.0	1
4	1.51742	13.27	3.62	1.24	73.08	0.55	8.07	0.0	0.0	1

In [7]:
```
df.shape
```

Out[7]:
```
(214, 10)
```

In [11]:
```
X = df.drop(['Type'], axis = 1)
y = df['Type']
```

In [217]:
```
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split funct

#splitting data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, random_stat

#Create Decision Tree classifer object (Information Gain)
clf = DecisionTreeClassifier(criterion = 'gini', splitter = 'random', max_depth =7,

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)
```

In [36]:
```
from sklearn import metrics #Import scikit-learn metrics module for accuracy calcula

#Predict the response for test dataset
y_pred = clf.predict(X_test)

print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.66
```

In [37]:
```
# Creating list of Accuracy scores for range of max depth values

k_range = range(1,40)
accuracy = []

for k in k_range:
    clf = DecisionTreeClassifier(criterion = 'entropy', splitter = 'random', max_dep
    clf = clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)
    accuracy_scores = metrics.accuracy_score(y_test, y_pred)
    accuracy.append(accuracy_scores)

print("Accuracy:", accuracy)
print("Max Accuracy Score:", max(accuracy))
print("Max_depth:", accuracy.index(max(accuracy)))
```

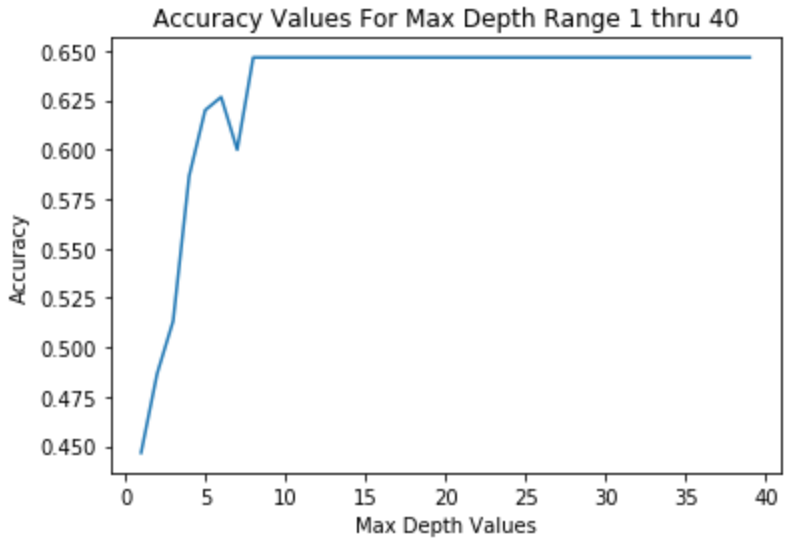
```
Accuracy: [0.4466666666666666, 0.4866666666666667, 0.5133333333333333, 0.58666666
66666667, 0.62, 0.6266666666666667, 0.6, 0.6466666666666666, 0.6466666666666666,
0.6466666666666666, 0.6466666666666666, 0.6466666666666666, 0.6466666666666666, 0.
6466666666666666, 0.6466666666666666, 0.6466666666666666, 0.6466666666666666, 0.64
66666666666666, 0.6466666666666666, 0.6466666666666666, 0.6466666666666666, 0.6466
666666666666, 0.6466666666666666, 0.6466666666666666, 0.6466666666666666, 0.646666
6666666666, 0.6466666666666666, 0.6466666666666666, 0.6466666666666666, 0.64666666
66666666, 0.6466666666666666, 0.6466666666666666, 0.6466666666666666, 0.6466666666
666666, 0.6466666666666666, 0.6466666666666666, 0.6466666666666666, 0.646666666666
6666, 0.6466666666666666]
Max Accuracy Score: 0.6466666666666666
Max_depth: 7
```

In [38]:
```
#Visualization of Accuracy scores vs. Max depth

import matplotlib.pyplot as plt
%matplotlib inline

plt.plot(k_range, accuracy)
plt.title('Accuracy Values For Max Depth Range 1 thru 40')
plt.xlabel('Max Depth Values')
plt.ylabel('Accuracy')
```

Out[38]:
```
Text(0, 0.5, 'Accuracy')
```



Using Ensemble Technique AdaBoost to Increase Accuracy of Decision Tree Classifier

In [356]:
```
from sklearn.ensemble import AdaBoostClassifier

# Create adaboost classifer object
abc = AdaBoostClassifier(clf, n_estimators=340,
                         learning_rate=1.75, algorithm="SAMME")

# Train Adaboost Classifer
model = abc.fit(X_train, y_train)
```

In [357]:
```
y_pred_abc = model.predict(X_test)
```

In [358]:
```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred_abc))
```

```
Accuracy: 0.7466666666666667
```

In [ ]:

In [ ]: