In [25]:

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn import metrics
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
import matplotlib.pyplot as plt
```

In [26]:

```
df = pd.read_csv(r"C:\Users\Sarthak Vora\Downloads\responses.csv")
X = df[df.columns[0:7]]
Y = df["Category"]
print(Y.shape)
print(X.shape)
```

```
(167,)
(167, 7)
```

In [27]:

```
LDA_transform = LDA(n_components=2)
LDA_transform.fit(X,Y)
X_reduced = LDA_transform.transform(X)
print("Original # features:",X.shape[1])
print("New # features:", X_reduced.shape[1])
```

```
Original # features: 7
New # features: 2
```

In [28]:

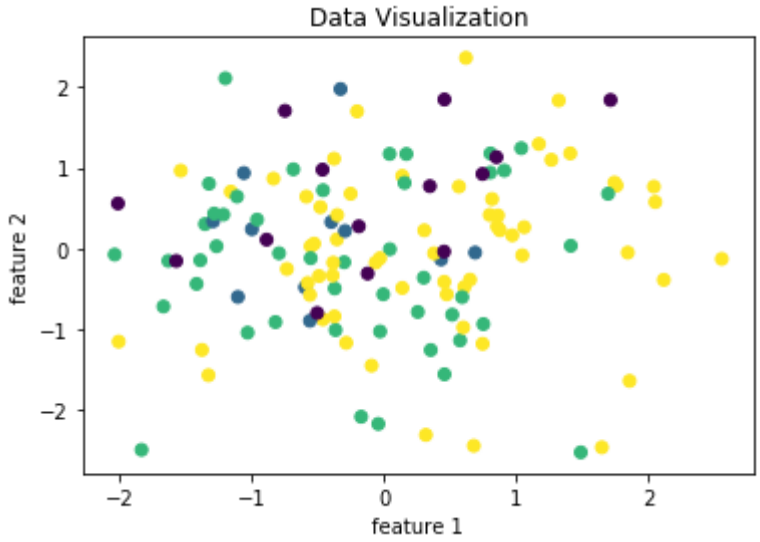
```
# Random Splitting of data to devset and test set with a test set size of 25%
X_train, X_test, Y_train, Y_test = train_test_split(X_reduced, Y, test_size=0.20, rando
m_state=320)
```

In [29]:

```
# Using Normalization by training it on devset
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

In [30]:

```
plt.scatter(X_train[:,0], X_train[:,1], c = Y_train)
plt.xlabel("feature 1")
plt.ylabel("feature 2")
plt.title("Data Visualization")
plt.show()
```



In [15]:

```
clf = SVC()
clf.fit(X_train, Y_train)
```

Out[15]:

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

In [16]:

```
y_pred = clf.predict(X_test)
print(np.array(Y_test))
print(np.array(y_pred))
```

```
[2 3 3 3 3 1 2 2 2 1 3 3 3 2 2 3 3 3 2 3 2 3 3 2 0 3 2 3 2 3 2 3 3 3]
[3 3 3 3 3 2 2 3 3 3 3 3 3 3 3 3 3 2 3 3 3 2 3 3 3 2 3 2 3 3 3 3 3]
```

In [17]:

```
accuracy = metrics.accuracy_score(Y_test,y_pred)
print(accuracy)
```

```
0.6470588235294118
```

In [ ]:

```

```

In [ ]: