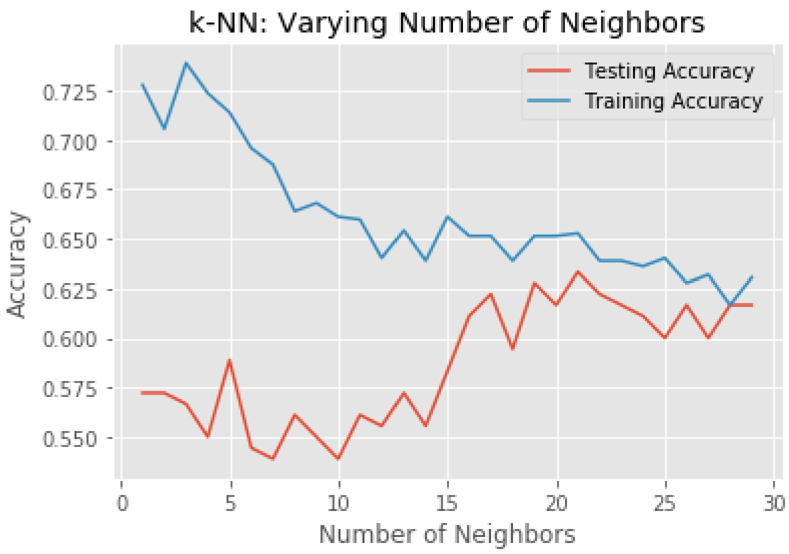
In [47]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
plt.style.use('ggplot')
df = pd.read_csv('file:///Users/mengyuanchen/Desktop/Treasury Squeeze test - DS1.csv', header=None)
df = df.values
y = df[1:,11]
X = df[1:,2:10]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state=10, stratify=y)

# cite： code is orgianl from Datacamp classification and classification from and regression tree
```

In [48]:

```
# KNN
#Setup arrays to store train and test accuracies
neighbors = np.arange(1, 30)
train_accuracy = np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))

# Loop over different values of k
for i, k in enumerate(neighbors):
    # Setup a k-NN Classifier with k neighbors: knn
    knn = KNeighborsClassifier(n_neighbors=k)

    # Fit the classifier to the trainng data
    knn.fit(X_train,y_train)

    #Compute accuracy on the training set
    train_accuracy[i] = knn.score(X_train, y_train)

    #Compute accuracy on the testing set
    test_accuracy[i] = knn.score(X_test, y_test)

# Generate plot
plt.title('k-NN: Varying Number of Neighbors')
plt.plot(neighbors, test_accuracy, label = 'Testing Accuracy')
plt.plot(neighbors, train_accuracy, label = 'Training Accuracy')
plt.legend()
plt.xlabel('Number of Neighbors')
plt.ylabel('Accuracy')
plt.show()
```



In [72]:

```
# Import DecisionTreeClassifier from sklearn.tree
from sklearn.tree import DecisionTreeClassifier


# Instantiate a DecisionTreeClassifier 'dt' with a maximum depth of 6
dt = DecisionTreeClassifier(max_depth=6, random_state=1)

# Fit dt to the training set
dt.fit(X_train, y_train)

# Predict test set labels
y_pred = dt.predict(X_test)
print(y_pred[0:26])

# Import accuracy_score
from sklearn.metrics import accuracy_score

# Predict test set labels
y_pred = dt.predict(X_test)

# Compute test set accuracy
acc = accuracy_score(y_test, y_pred)
print("Test set accuracy: {:.2f}".format(acc))
```

```
['FALSE' 'FALSE' 'FALSE' 'FALSE' 'TRUE' 'FALSE' 'FALSE' 'TRUE' 'TRU
E'
 'TRUE' 'FALSE' 'TRUE' 'FALSE' 'FALSE' 'FALSE' 'FALSE' 'FALSE' 'FALS
E'
 'FALSE' 'TRUE' 'FALSE' 'FALSE' 'FALSE' 'FALSE' 'FALSE' 'FALSE']
Test set accuracy: 0.58
```

In [74]:

```
# Import DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier
# Import train_test_split
from sklearn.model_selection import train_test_split
# Import accuracy_score
from sklearn.metrics import accuracy_score
# Split dataset into 80% train, 20% test
X_train, X_test, y_train, y_test= train_test_split(X, y,
test_size=0.2,stratify=y,random_state=10)
# Instantiate dt, set 'criterion' to 'gini'
dt = DecisionTreeClassifier(criterion='gini', random_state=1)

# Fit dt to the training set
dt.fit(X_train,y_train)
# Predict test-set labels
y_pred= dt.predict(X_test)
# Evaluate test-set accuracy
accuracy_score(y_test, y_pred)
```

Out[74]:

```
0.5722222222222222
```

In [63]:

```
print("My name is {Mengyuan Chen}")
print("My NetID is: {mchen100}")
print("I hereby certify that I have read the University policy on Academic Integrity and that I am not in violation.")
```

In [ ]:

```
```