Applying Decision Tree Classifier on DonorsChoose Dataset with TFIDF weighted Word 2 Vec and plotting the results.

```
%cd /content/drive/
%cd MyDrive/assignment/DonorsChoose/
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decision_tree import DecisionTreeClassifier
from sklearn.metrics import roc_auc_score,roc_curve,auc
from sklearn.metrics import confusion_matrix
from sklearn.metrics import *
```

```
[Warning: Cannot change to, unknown directory: /content/drive/
/content/drive
[Warning: Cannot change to, unknown directory: MyDrive/assignment/DonorsChoose/
/content/drive/MyDrive/assignment/DonorsChoose
```

```
%cd /content/drive/MyDrive/assignment/DonorsChoose/
!ls -ltrh train_data.csv cv_data.csv test_data.csv train_data_preprocessed.pickle cv_data_preprocessed.pickle test_data_preprocessed.pickle
```

```
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 train_data.csv
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 cv_data.csv
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 test_data.csv
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 train_data_preprocessed.pickle
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 cv_data_preprocessed.pickle
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 test_data_preprocessed.pickle
```

```
/content/drive/MyDrive/assignment/DonorsChoose
```

```
project_grade_category teacher_prefix school_state teacher_number_of_previously_posted_projects  train_categories  train_sub_categories
0   Grades PreK-2    Mrs.       CA                                 53   Literacy_Language  ESL_Literacy
1   Grades 3-5       Ms.        UT                                 4    History_Civics     Civics_Government
```

TFIDF approach for feature 'essay'

```
%cd /content/drive/MyDrive/assignment/DonorsChoose/
train_essay_tfidf_w2v = pickle.load(open('train_essay_tfidf_w2v.pickle', 'rb'))
cv_essay_tfidf_w2v = pickle.load(open('cv_essay_tfidf_w2v.pickle', 'rb'))
test_essay_tfidf_w2v = pickle.load(open('test_essay_tfidf_w2v.pickle', 'rb'))
```

```
/content/drive/MyDrive/assignment/DonorsChoose
```

```
%cd /content/drive/MyDrive/assignment/DonorsChoose/
!ls -ltrh train_data.csv cv_data.csv test_data.csv train_data_preprocessed.pickle cv_data_preprocessed.pickle test_data_preprocessed.pickle train_essay_tfidf_w2v.pickle cv_essay_tfidf_w2v.pickle test_essay_tfidf_w2v.pickle
```

```
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 train_data.csv
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 cv_data.csv
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 test_data.csv
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 train_data_preprocessed.pickle
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 cv_data_preprocessed.pickle
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 test_data_preprocessed.pickle
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 train_essay_tfidf_w2v.pickle
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 cv_essay_tfidf_w2v.pickle
-rw-rw-rw- 1 root root 1.1G Jun  2 05:37 test_essay_tfidf_w2v.pickle
```

```
# Find the indexes of numerical and category features
numerical_indexes = [i for i in range(train_essay_tfidf_w2v.shape[1])]
print("numerical_indexes",numerical_indexes)
print("number of numerical features",len(numerical_indexes))
print("number of numerical features",train_essay_tfidf_w2v.shape[1])
print("number of numerical features",cv_essay_tfidf_w2v.shape[1])
print("number of numerical features",test_essay_tfidf_w2v.shape[1])
```

```
numerical_indexes [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
```

```
# Plot ROC curve
plot_roc_curve(model, test_data[,-c(1:2)], test_data[,c(2)])
```



```
# Calculate AUC
calc_auc(model, test_data[,-c(1:2)], test_data[,c(2)])
```

```
# Plotting model with best hyperparameters
plot_learning_curve(model, train_data[,-c(1:2)], train_data[,c(2)])
```

```
# Plotting model with best hyperparameters
plot_learning_curve(model, train_data[,-c(1:2)], train_data[,c(2)])
```

```
# Plotting ROC curve with best hyperparameters
plot_roc_curve(model, test_data[,-c(1:2)], test_data[,c(2)])
```



```
# Calculate AUC
calc_auc(model, test_data[,-c(1:2)], test_data[,c(2)])
```