N-R Team

E Navaneet Kumar, Rithvika Paladugu

In [1]:

```
import os
import numpy as np
import pandas as pd
```

In [2]:

```
FILES_DIR = 'C:/Users/navne/Python Files/Data Acquisition AAI 627/HW10_1'
ensemble_files = [
    {'file': 'random_forest_final_predictions_0859.csv', 'score': 0.859},
    {'file': 'decision_tree_final_predictions_0859.csv', 'score': 0.859},
    {'file': 'grad_boost_final_predictions_0863.csv', 'score': 0.863},
    {'file': 'log_reg_genre_final_predictions_0869.csv', 'score': 0.869},
    {'file': 'dt_final_predictions_0823.csv', 'score': 0.823},
    {'file': 'gbt_final_predictions_0844.csv', 'score': 0.844},
    {'file': 'lr_final_predictions_0845.csv', 'score': 0.845},
    {'file': 'rf_final_predictions_0823.csv', 'score': 0.823}
]
```

In [3]:

```
# Load sample submission to ensure alignment
df= pd.read_csv('sample_submission.csv').sort_values('TrackID').reset_index(drop=True)
```

In [4]:

```
scores = []
predictions = []
for ensemble_file in ensemble_files:
    pred_df = pd.read_csv(os.path.join(FILES_DIR, ensemble_file['file'])).sort_values('T
rackID')
    # Ensure that pred_df aligns with df_sample in terms of TrackIDs
    aligned_pred = df[['TrackID']].merge(pred_df, on='TrackID', how='left')['Predictor']
.fillna(0).values  # Fill NA with 0 or other imputation method
    predictions.append(aligned_pred)
    scores.append(ensemble_file['score'])

    print(f"Loaded {ensemble_file['file']} with shape {aligned_pred.shape}")
```

```
Loaded random_forest_final_predictions_0859.csv with shape (120000,)
Loaded decision_tree_final_predictions_0859.csv with shape (120000,)
Loaded grad_boost_final_predictions_0863.csv with shape (120000,)
Loaded log_reg_genre_final_predictions_0869.csv with shape (120000,)
Loaded dt_final_predictions_0823.csv with shape (120000,)
Loaded gbt_final_predictions_0844.csv with shape (120000,)
Loaded lr_final_predictions_0845.csv with shape (120000,)
Loaded rf_final_predictions_0823.csv with shape (120000,)
```

In [5]:

```
# Check if all predictions arrays have the same shape
print("All predictions have the same shape:", all(pred.shape == predictions[0].shape for
pred in predictions))

# If they all have the same shape, proceed to stack
if all(pred.shape == predictions[0].shape for pred in predictions):
    S = np.stack(predictions).T * 2 - 1
    print("Successfully created matrix S")
else:
    print("Error: Not all prediction arrays have the same shape.")
```

```
All predictions have the same shape: True
Successfully created matrix S
```

In [6]:

```
StS_inv_pseudo = np.linalg.pinv(S.T.dot(S))

StX = len(S)*(np.array(scores)*2 - 1)
```

In [7]:

```
a_LS = StS_inv_pseudo.dot(StX)
```

In [8]:

```
df['EnsembleScore'] = S.dot(np.expand_dims(a_LS, axis=-1))
```

In [9]:

```
df['UserID'] = df['TrackID'].str.split('_').str[0]
```

In [10]:

```
df = df.sort_values('EnsembleScore', ascending=False)
```

In [11]:

```
df = pd.concat([
    df.groupby(['UserID']).head(3).assign(Predictor=1),
    df.groupby(['UserID']).tail(3).assign(Predictor=0)
])[['TrackID', 'Predictor']]
```

In [12]:

```
df.to_csv('ensemble_final.csv', index=None)
```

The best public score after ensembling method we got is 0.869 which is better than before and put us at the 8th position in the Kaggle leaderboard.

In [ ]:

```

```