In [97]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pyreadstat

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.impute import SimpleImputer
from sklearn.metrics import precision_score, recall_score, f1_score
```

In [55]:
```
# calling rh and lh volumes
left_volume_file_path = r"Z:\Active-Diagnose_CTE\Fargol_Analysis\Volumetric_Analysis\lh_aparc_volume_2023111
left_volume = pd.read_csv(left_volume_file_path)
left_volume = pd.DataFrame(left_volume)

right_volume_file_path = r"Z:\Active-Diagnose_CTE\Fargol_Analysis\Volumetric_Analysis\rh_aparc_volume_202311
right_volume = pd.read_csv(right_volume_file_path)
right_volume = pd.DataFrame(right_volume)
```

In [56]:
```
left_volume.head()
right_volume.head()
```

Out[56]:

	subject_id	visit	checkin_bin	exposurebin	age_decade	racecat_combined	eduyears	totyr_foot	chiiseas_pf	chiiyrs_pf	...	rh_ros
0	1001	1	2	1	1	5	16.0	7.0	4335.4	2167.7	...	
1	1002	1	2	1	1	5	15.0	14.0	10363.1	5708.1	...	
2	1003	1	2	1	1	5	18.0	12.0	6685.4	4863.9	...	
3	1004	1	1	1	2	5	16.0	16.0	7701.2	6448.9	...	
4	1005	1	3	0	2	5	21.0	NaN	NaN	NaN	...	

5 rows × 51 columns

In [57]:
```
print("Column Names:")
print(right_volume.columns[2])
```

```
Column Names:
checkin_bin
```

In [74]:
```
#group them base on the value in the third column which indicates their level of playing
right_grouped = right_volume.groupby(right_volume.iloc[:, 2])
left_grouped = left_volume.groupby(left_volume.iloc[:, 2])

NFL_right_grouped = pd.DataFrame()
CP_right_grouped = pd.DataFrame()
HC_right_grouped = pd.DataFrame()


# group_name : 1, 2, 3   group_data:
for group_name, group_data in right_grouped:
    if group_name == 1:
        NFL_right_grouped = pd.concat([NFL_right_grouped,group_data], ignore_index = True)
    if group_name == 2:
        CP_right_grouped = pd.concat([CP_right_grouped,group_data], ignore_index = True)
    if group_name == 3:
        HC_right_grouped = pd.concat([HC_right_grouped,group_data], ignore_index = True)

#print("DataFrame for NFL:")
#print(NFL_right_grouped.head())
```

In [75]:
```
NFL_right_grouped.head()
#print(NFL_right_grouped.columns)
```

Out[75]:

	subject_id	visit	checkin_bin	exposurebin	age_decade	racecat_combined	eduyears	totyr_foot	chiiseas_pf	chiiyrs_pf	...	rh_ros
0	1004	1	1	1	2	5	16.0	16.0	7701.2	6448.9	...	
1	1008	1	1	1	2	3	15.0	22.0	8220.9	5421.2	...	
2	1011	1	1	1	2	5	16.0	20.0	9307.0	9307.0	...	
3	1015	1	1	1	1	3	19.0	17.0	9866.7	6173.3	...	
4	1018	1	1	1	1	3	16.0	23.0	10635.9	7929.6	...	

5 rows × 51 columns

In [76]:
```
#NFL_right_grouped.columns[[1] +list(range(3,index_of_Atlas+1))]
CP_right_grouped.columns[[1] +list(range(3,index_of_Atlas+1))]
```

Out[76]:
```
Index(['visit', 'exposurebin', 'age_decade', 'racecat_combined', 'eduyears',
       'totyr_foot', 'chiiseas_pf', 'chiiyrs_pf', 'chiiseas_pl', 'chiiyrs_pl',
       'chiiseas_pg', 'chiiyrs_pg', 'timepoint_aparc', 'FreeSurfer_Version',
       'Atlas'],
      dtype='object')
```

In [77]:
```
# Atlas is the last column that needs to be deleted
index_of_Atlas = NFL_right_grouped.columns.get_loc("Atlas")
NFL_right_grouped.drop(columns=NFL_right_grouped.columns[[1] +list(range(3,index_of_Atlas+1))], inplace = Tr
CP_right_grouped.drop(columns=CP_right_grouped.columns[[1] +list(range(3,index_of_Atlas+1))], inplace = True
HC_right_grouped.drop(columns=HC_right_grouped.columns[[1] +list(range(3,index_of_Atlas+1))], inplace = True
```

In [78]:
```
NFL_right_grouped.head()
```

Out[78]:

	subject_id	checkin_bin	rh_bankssts_volume	rh_caudalanteriorcingulate_volum	rh_caudalmiddlefrontal_volume	rh_cuneus_volume	r
0	1004	1	2310.0	1647.0	4656.0	3471.0	
1	1008	1	1946.0	1687.0	4961.0	3116.0	
2	1011	1	1961.0	2483.0	6019.0	4356.0	
3	1015	1	2092.0	2032.0	5290.0	3845.0	
4	1018	1	2547.0	2028.0	5994.0	4281.0	

5 rows × 36 columns

In [80]:
```
#combine all three classes
combined_right_volume = pd.concat([NFL_right_grouped, CP_right_grouped, HC_right_grouped], ignore_index=True
```

In [81]:
```
combined_right_volume.head()
```

Out[81]:

	subject_id	checkin_bin	rh_bankssts_volume	rh_caudalanteriorcingulate_volum	rh_caudalmiddlefrontal_volume	rh_cuneus_volume	r
0	1004	1	2310.0	1647.0	4656.0	3471.0	
1	1008	1	1946.0	1687.0	4961.0	3116.0	
2	1011	1	1961.0	2483.0	6019.0	4356.0	
3	1015	1	2092.0	2032.0	5290.0	3845.0	
4	1018	1	2547.0	2028.0	5994.0	4281.0	

5 rows × 36 columns

In [82]:
```
# Separate based on the level of professionalism
X = combined_right_volume.drop(columns='checkin_bin')  # Adjust 'Label' to the actual column name containing
y = combined_right_volume['checkin_bin']
```

In [83]:
```
# Splitting
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [88]:
```
# Normalization
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [91]:
```
# Replace NaNs with means
imputer = SimpleImputer(strategy='mean')  # You can choose a different strategy
X_train_imputed = imputer.fit_transform(X_train_scaled)
X_test_imputed = imputer.transform(X_test_scaled)
```

In [ ]:
```
```

In [92]:
```
# train the model
model = RandomForestClassifier()
model.fit(X_train_imputed, y_train)
```

Out[92]:
```
▾ RandomForestClassifier

RandomForestClassifier()
```

In [93]:
```
# prediction
y_pred = model.predict(X_test_imputed)
```

In [98]:
```
# Evaluation
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')
print(f"Accuracy: {accuracy}, Precision: {precision}, Recall: {recall}, F1-Score: {f1}")
```

```
Accuracy: 0.4583333333333333, Precision: 0.38715277777777773, Recall: 0.4583333333333333, F1-Score: 0.37915
80667354661
```

In [ ]:
```
```