In [15]:
```
#Import Libraries and Load Data
import pandas as pd
import numpy as np
from sklearn.model_selection import StratifiedKFold, train_test_split
from catboost import CatBoostClassifier
from sklearn.metrics import roc_auc_score
import optuna
import warnings
import matplotlib.pyplot as plt
import seaborn as sns

warnings.filterwarnings('ignore')
```

In [16]:
```
#Exploratory Data Analysis (EDA)
# Display basic information about the datasets
print("Train Data Info:")
print(train_df.info())
print("\nTest Data Info:")
print(test_df.info())

# Display the first few rows of the datasets
print("\nTrain Data Head:")
print(train_df.head())
print("\nTest Data Head:")
print(test_df.head())

# Check for missing values
print("\nMissing Values in Train Data:")
print(train_df.isnull().sum())
print("\nMissing Values in Test Data:")
print(test_df.isnull().sum())

# Basic statistics of the datasets
print("\nTrain Data Description:")
print(train_df.describe())
print("\nTest Data Description:")
print(test_df.describe())
```

```
Train Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29675 entries, 0 to 29674
Columns: 247 entries, game_time to ID
dtypes: bool(1), float64(30), int64(215), object(1)
memory usage: 55.7+ MB
None

Test Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Columns: 246 entries, game_time to ID
dtypes: float64(30), int64(215), object(1)
memory usage: 18.8+ MB
None

Train Data Head:
   game_time  game_mode  lobby_type  objectives_len  chat_len  r1_hero_id  \
0        871         22           0               4         2         110
1       2549         22           0              17         0         114
2       1841         22           0               8         1         100
3       2211         22           7              11         3          32
4        458         22           7               1         0          68

   r1_kills  r1_deaths  r1_assists  r1_denies  ...  d5_camps_stacked  \
0         2          3          11          3  ...                 0
1        16          2          12         24  ...                 1
2         2         11          12          2  ...                 0
3        14          3          11         21  ...                 0
4         3          0           0         15  ...                 0

   d5_rune_pickups  d5_firstblood_claimed  d5_teamfight_participation  \
0               10                      1                    0.875000
1               10                      0                    0.535714
2               13                      0                    0.727273
3                1                      0                    0.347826
4                6                      0                    0.000000

   d5_towers_killed  d5_roshans_killed  d5_obs_placed  d5_sen_placed  \
0                 0                  0              6              3
1                 1                  0              0              0
2                 0                  0              0              0
3                 0                  0              0              0
4                 0                  0              1              1

   radiant_win                                ID
0         True  a363534a6344f1b0be1d7ba2c4047d9a
1         True  a0ba4ef0965f56d2eba69c2b9ef33353
2         True  18873e56c2142af326b4e08ca41df63a
3         True  c143931a6a8b3fb55a8ef6b9f30c6933
4         True  5a324d8b37522e9f9684493465720023

[5 rows x 247 columns]

Test Data Head:
   game_time  game_mode  lobby_type  objectives_len  chat_len  r1_hero_id  \
0        155         22           7               1        11          11
1       1362          4           0               6         4          39
2       2388          4           0              16        10         103
3       2043         22           0              15        49          44
4        840         22           7               2        27           9

   r1_kills  r1_deaths  r1_assists  r1_denies  ...  d5_creeps_stacked  \
0         0          0           0          0  ...                  0
1         1          1           4         13  ...                  9
2         9          8          18          3  ...                  0
```

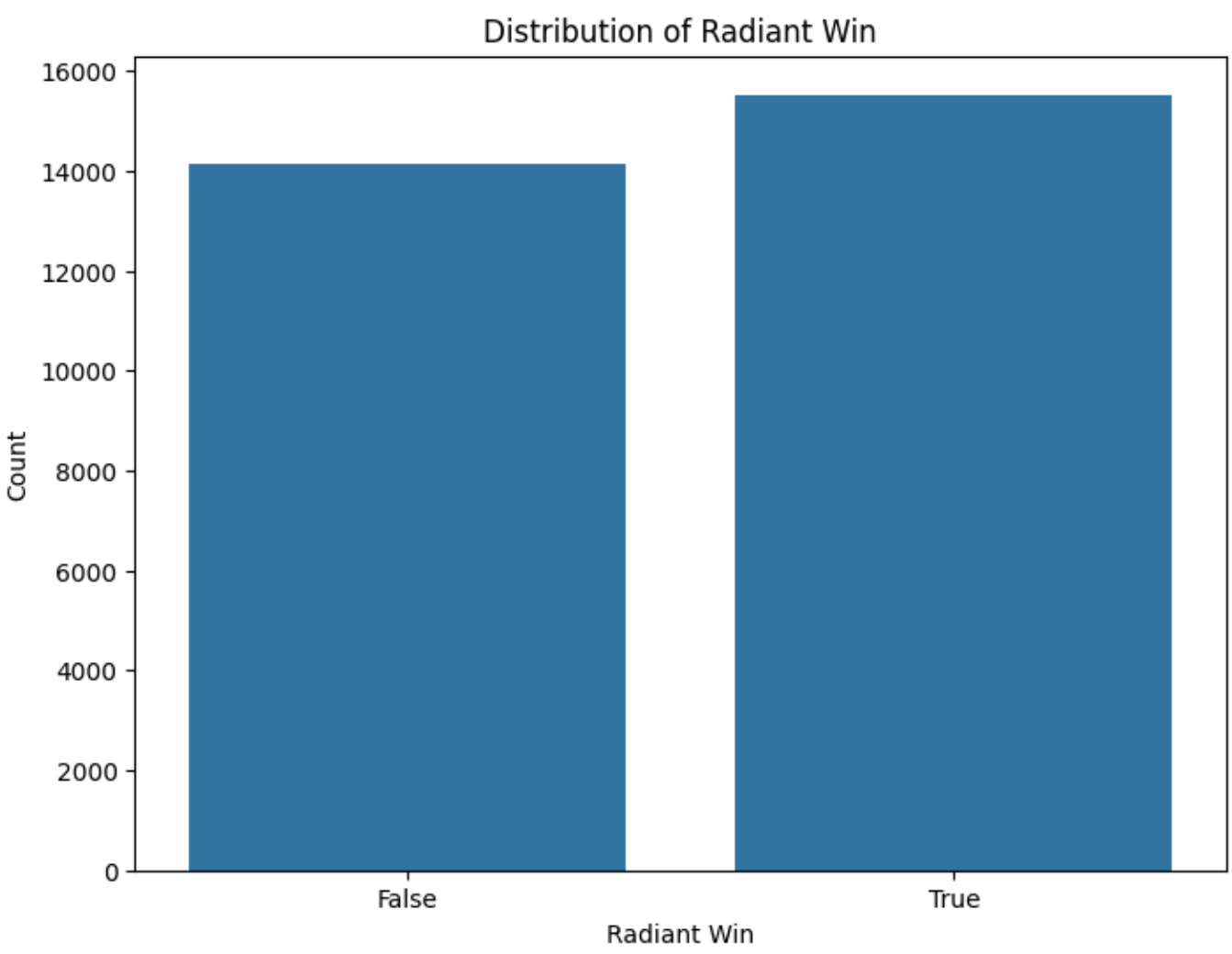
```
3         3          6           3          7  ...                 0
4         1          2           4          2  ...                 0

   d5_camps_stacked  d5_rune_pickups  d5_firstblood_claimed  \
0                 0                0                      0
1                 3               13                      0
2                 0                6                      0
3                 0                4                      0
4                 0                1                      0

   d5_teamfight_participation  d5_towers_killed  d5_roshans_killed  \
0                    0.000000                 0                  0
1                    0.222222                 0                  0
2                    0.686275                 2                  0
3                    0.945946                 2                  0
4                    0.375000                 1                  0

   d5_obs_placed  d5_sen_placed                                ID
0              0              0  a400b8f29dece5f4d266f49f1ae2e98a
1              0              0  34c81a8faede0d8f1f87dcc6ee824658
2              0              0  5feece770ca79e5e8cd8052198b3f533
3              1              0  8f56cc2468ba5c37edb79f3a7b4af6e6
4              0              0  44cdded6d3311134563f743eb77685b2

[5 rows x 246 columns]

Missing Values in Train Data:
game_time            0
game_mode            0
lobby_type           0
objectives_len       0
chat_len             0
                   ..
d5_roshans_killed    0
d5_obs_placed        0
d5_sen_placed        0
radiant_win          0
ID                   0
Length: 247, dtype: int64

Missing Values in Test Data:
game_time            0
game_mode            0
lobby_type           0
objectives_len       0
chat_len             0
                   ..
d5_towers_killed     0
d5_roshans_killed    0
d5_obs_placed        0
d5_sen_placed        0
ID                   0
Length: 246, dtype: int64

Train Data Description:
          game_time     game_mode    lobby_type  objectives_len      chat_len  \
count  29675.000000  29675.000000  29675.000000    29675.000000  29675.000000
mean    1145.695333     19.592957      4.772738        6.531997      7.351879
std      768.974419      6.297211      3.260444        6.500606     13.561608
min        0.000000      2.000000      0.000000        0.000000      0.000000
25%      518.000000     22.000000      0.000000        1.000000      0.000000
50%     1043.000000     22.000000      7.000000        4.000000      3.000000
75%     1654.000000     22.000000      7.000000       10.000000      9.000000
max     4742.000000     23.000000      7.000000       43.000000    291.000000

         r1_hero_id      r1_kills     r1_deaths    r1_assists     r1_denies  \
```

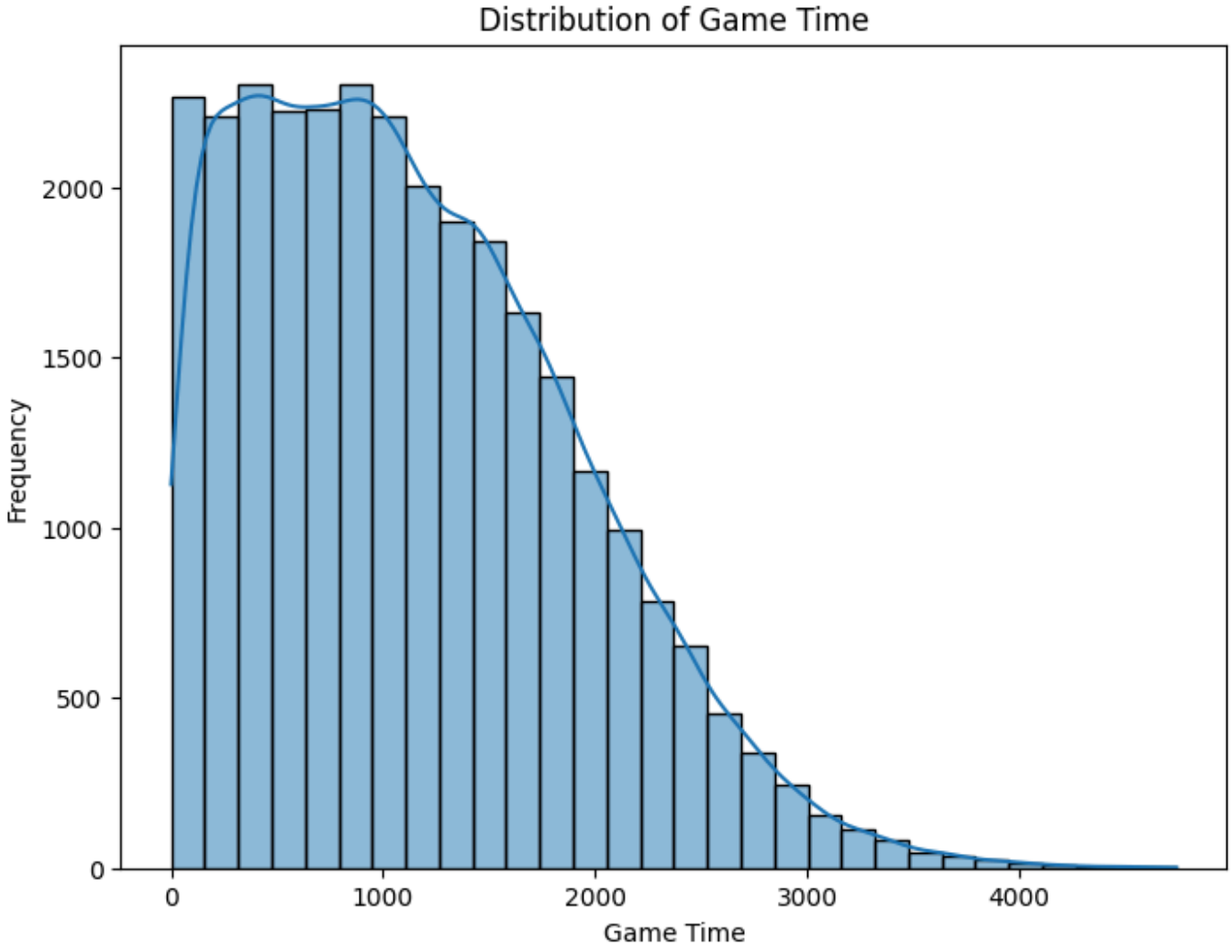
```
count  29675.000000  29675.000000  29675.000000  29675.000000  29675.000000
mean      51.103960      3.155619      3.256917      4.666992      6.335771
std       34.590915      3.744580      3.279818      5.225634      8.273026
min        1.000000      0.000000      0.000000      0.000000      0.000000
25%       20.000000      0.000000      1.000000      1.000000      1.000000
50%       44.000000      2.000000      2.000000      3.000000      3.000000
75%       81.000000      5.000000      5.000000      7.000000      9.000000
max      120.000000     32.000000     27.000000     40.000000     84.000000

       ...      d5_stuns  d5_creeps_stacked  d5_camps_stacked  \
count  ...  29675.000000       29675.000000      29675.000000
mean   ...     11.722391           1.038585          0.341769
std    ...     20.658919           3.556953          0.962948
min    ...     -6.191284           0.000000          0.000000
25%    ...      0.000000           0.000000          0.000000
50%    ...      1.366956           0.000000          0.000000
75%    ...     15.863060           0.000000          0.000000
max    ...    277.618070         132.000000         29.000000

       d5_rune_pickups  d5_firstblood_claimed  d5_teamfight_participation  \
count     29675.000000           29675.000000                29675.000000
mean          4.694457               0.088795                    0.414476
std           4.645499               0.284453                    0.267860
min           0.000000               0.000000                    0.000000
25%           1.000000               0.000000                    0.230769
50%           3.000000               0.000000                    0.440000
75%           7.000000               0.000000                    0.600000
max          57.000000               1.000000                    2.000000

       d5_towers_killed  d5_roshans_killed  d5_obs_placed  d5_sen_placed
count      29675.000000       29675.000000   29675.000000   29675.000000
mean           0.302072           0.024971       1.276765       0.789250
std            0.740563           0.170487       2.602252       2.448526
min            0.000000           0.000000       0.000000       0.000000
25%            0.000000           0.000000       0.000000       0.000000
50%            0.000000           0.000000       0.000000       0.000000
75%            0.000000           0.000000       1.000000       0.000000
max            9.000000           4.000000      26.000000      47.000000

[8 rows x 245 columns]

Test Data Description:
          game_time     game_mode    lobby_type  objectives_len     chat_len  \
count  10000.000000  10000.000000  10000.000000    10000.000000  10000.00000
mean    1147.232600     19.560500      4.771200        6.503700      7.29880
std      761.973655      6.328218      3.261152        6.467099     12.77006
min        0.000000      2.000000      0.000000        0.000000      0.00000
25%      528.000000     22.000000      0.000000        1.000000      0.00000
50%     1048.500000     22.000000      7.000000        4.000000      3.00000
75%     1660.000000     22.000000      7.000000       10.000000      9.00000
max     4933.000000     23.000000      7.000000       41.000000    190.00000

         r1_hero_id      r1_kills     r1_deaths    r1_assists     r1_denies  \
count  10000.000000  10000.000000  10000.000000  10000.000000  10000.000000
mean      51.100700      3.124900      3.304100      4.679600      6.152700
std       34.640793      3.663474      3.293616      5.224756      7.994321
min        1.000000      0.000000      0.000000      0.000000      0.000000
25%       20.000000      0.000000      1.000000      1.000000      1.000000
50%       44.000000      2.000000      2.000000      3.000000      3.000000
75%       81.000000      5.000000      5.000000      7.000000      8.000000
max      120.000000     31.000000     25.000000     34.000000     70.000000

       ...      d5_stuns  d5_creeps_stacked  d5_camps_stacked  \
count  ...  10000.000000       10000.000000      10000.000000
mean   ...     11.924468           1.040300          0.347200
std    ...     20.337541           3.466393          0.966098
```

```
min    ...     -3.274992          0.000000          0.000000
25%    ...      0.000000          0.000000          0.000000
50%    ...      1.766235          0.000000          0.000000
75%    ...     16.462201          0.000000          0.000000
max    ...    235.240020         75.000000         17.000000

       d5_rune_pickups  d5_firstblood_claimed  d5_teamfight_participation  \
count     10000.000000           10000.000000                10000.000000
mean          4.652600               0.094100                    0.420367
std           4.636536               0.291982                    0.266596
min           0.000000               0.000000                    0.000000
25%           1.000000               0.000000                    0.250000
50%           3.000000               0.000000                    0.444444
75%           7.000000               0.000000                    0.600000
max          42.000000               1.000000                    2.000000

       d5_towers_killed  d5_roshans_killed  d5_obs_placed  d5_sen_placed
count      10000.000000       10000.000000   10000.000000   10000.000000
mean           0.293100           0.022800       1.247100       0.765600
std            0.707985           0.170538       2.583546       2.406332
min            0.000000           0.000000       0.000000       0.000000
25%            0.000000           0.000000       0.000000       0.000000
50%            0.000000           0.000000       0.000000       0.000000
75%            0.000000           0.000000       1.000000       0.000000
max            7.000000           5.000000      25.000000      31.000000

[8 rows x 245 columns]
```

In [17]:

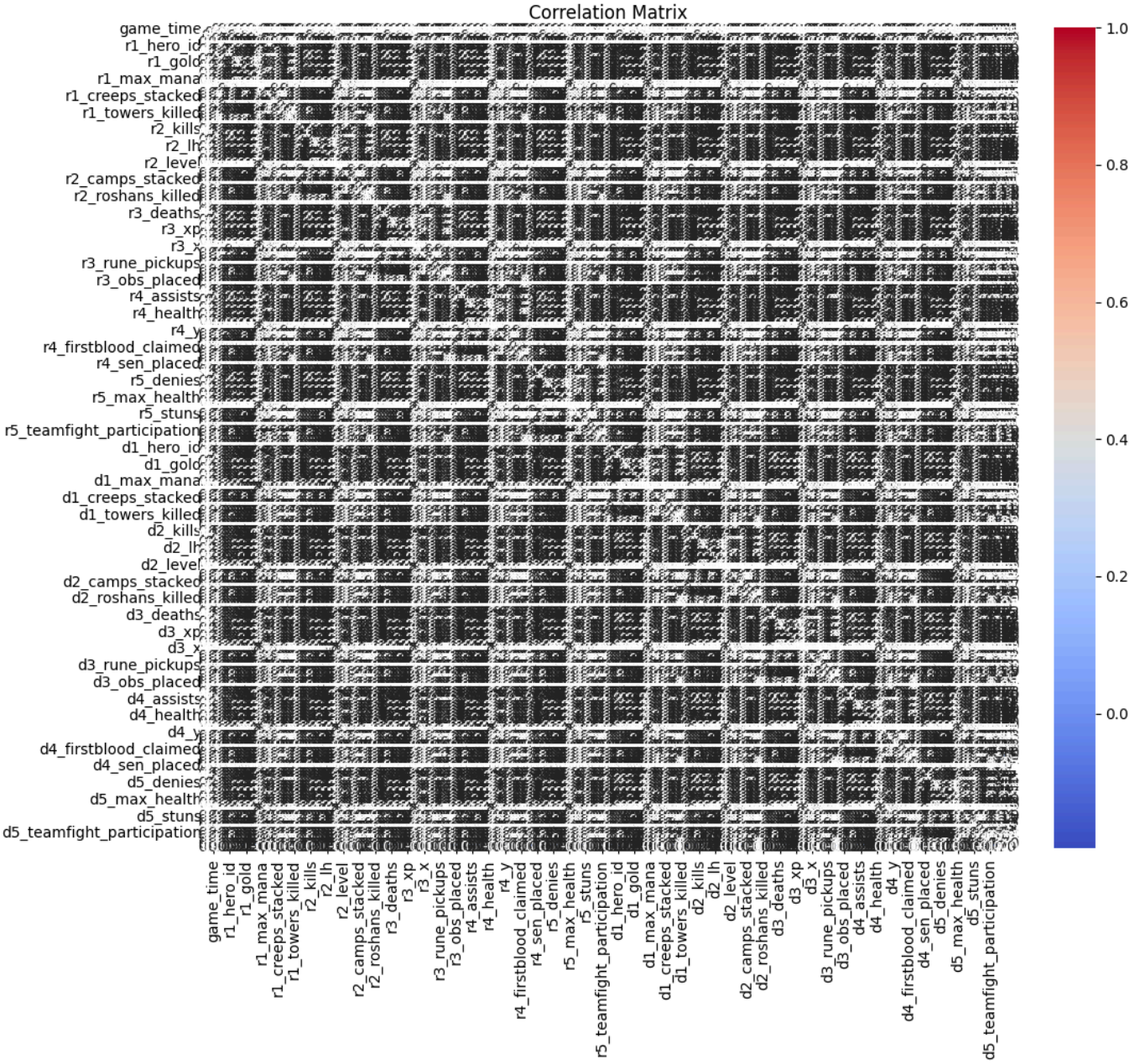
```python
#Distribution of Target Variable and Game Time

# Plot the distribution of the target variable 'radiant_win'
plt.figure(figsize=(8, 6))
sns.countplot(x='radiant_win', data=train_df)
plt.title('Distribution of Radiant Win')
plt.xlabel('Radiant Win')
plt.ylabel('Count')
plt.show()

# Plot the distribution of game time
plt.figure(figsize=(8, 6))
sns.histplot(train_df['game_time'], bins=30, kde=True)
plt.title('Distribution of Game Time')
plt.xlabel('Game Time')
plt.ylabel('Frequency')
plt.show()
```





In [19]:
```
#Correlation Matrix
```

```
numeric_cols = train_df.select_dtypes(include=[np.number]).columns.tolist()
correlation_matrix = train_df[numeric_cols].corr()

# Plot the correlation matrix
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```

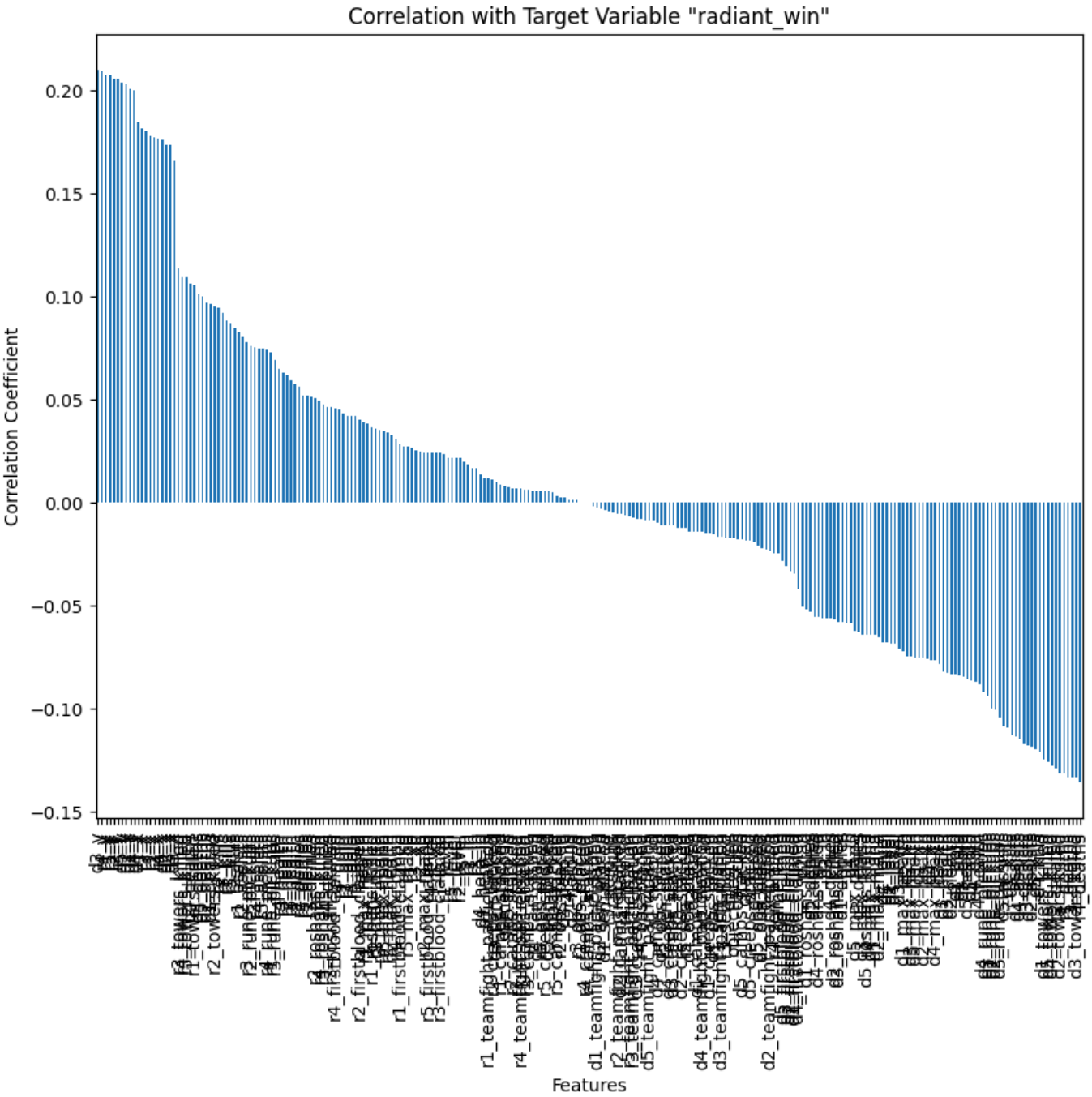


In [23]:

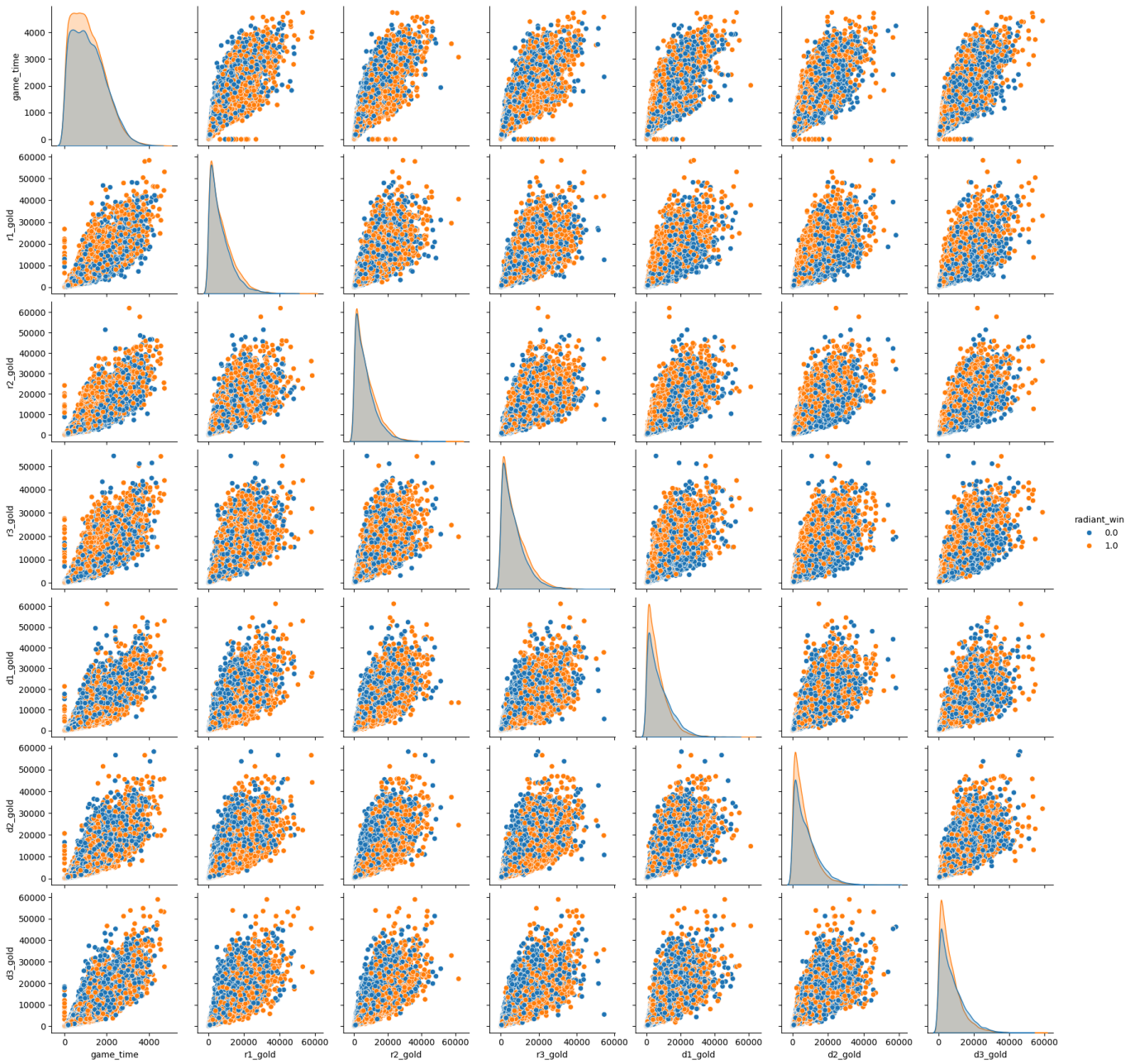
```
# Ensure 'radiant_win' is treated as a numeric column
train_df['radiant_win'] = train_df['radiant_win'].astype(float)

# Ensure all columns are numeric for correlation matrix calculation
numeric_cols = train_df.select_dtypes(include=[np.number]).columns.tolist()
correlation_matrix = train_df[numeric_cols].corr()

# Correlation with target variable 'radiant_win'
if 'radiant_win' in correlation_matrix.columns:
    correlation_with_target = correlation_matrix['radiant_win'].sort_values(ascending
    print("\nCorrelation with Target Variable 'radiant_win':")
    print(correlation_with_target)

    # Plot correlation with target variable 'radiant_win'
    plt.figure(figsize=(10, 8))
    correlation_with_target.drop('radiant_win').plot(kind='bar')
    plt.title('Correlation with Target Variable "radiant_win"')
    plt.xlabel('Features')
```

```
    plt.ylabel('Correlation Coefficient')
    plt.show()
else:
    print("The target variable 'radiant_win' is not present in the correlation matrix
```

```
Correlation with Target Variable 'radiant_win':
radiant_win         1.000000
d3_y                0.210176
r4_y                0.209289
r1_y                0.207705
r5_y                0.207584
                      ...
r1_deaths          -0.131264
r5_deaths          -0.133362
r3_deaths          -0.133410
d3_towers_killed   -0.133541
r4_deaths          -0.135563
Name: radiant_win, Length: 246, dtype: float64
```



In [24]:
```
# Pairplot for selected features
selected_features = ['game_time', 'r1_gold', 'r2_gold', 'r3_gold', 'd1_gold', 'd2_gol
sns.pairplot(train_df[selected_features], hue='radiant_win', diag_kind='kde')
plt.show()
```



In [25]:
```python
#Precompute Hero Win Rates
def compute_hero_win_rates(train_df):
    hero_stats = {}
    for _, row in train_df.iterrows():
        radiant_win = row['radiant_win']
        for team in ['r', 'd']:
            for pos in range(1, 6):
                hero_id = row[f'{team}{pos}_hero_id']
                win = radiant_win if team == 'r' else not radiant_win
                if hero_id not in hero_stats:
                    hero_stats[hero_id] = {'wins': 0, 'total': 0}
                hero_stats[hero_id]['wins'] += int(win)
                hero_stats[hero_id]['total'] += 1

    hero_win_rates = {hid: stats['wins']/stats['total'] for hid, stats in hero_stats.
    return hero_win_rates

hero_win_rates = compute_hero_win_rates(train_df)
```

In [26]:
```python
#Create Enhanced Features
def create_enhanced_features(df):
    features = {}
    game_time = df['game_time']

    # Time decay factors
    early_weight = np.exp(-game_time/900)
```

```
    mid_weight = np.exp(-game_time/1800)

    for team in ['r', 'd']:
        # Core positions (1-3)
        core_gold = df[[f'{team}{i}_gold' for i in range(1,4)]].sum(axis=1)
        core_xp = df[[f'{team}{i}_xp' for i in range(1,4)]].sum(axis=1)

        # Support positions (4-5)
        support_gold = df[[f'{team}{i}_gold' for i in range(4,6)]].sum(axis=1)
        support_xp = df[[f'{team}{i}_xp' for i in range(4,6)]].sum(axis=1)

        # Ultra Early Game Metrics
        features[f'{team}_core_early_gold'] = (core_gold / game_time) * early_weight
        features[f'{team}_core_early_xp'] = (core_xp / game_time) * early_weight * 2.
        features[f'{team}_support_early_gold'] = (support_gold / game_time) * early_w
        features[f'{team}_support_early_xp'] = (support_xp / game_time) * early_weigh

        # Mid Game Metrics
        total_gold = core_gold + support_gold
        total_xp = core_xp + support_xp
        features[f'{team}_mid_gold'] = (total_gold / game_time) * mid_weight * 1.2
        features[f'{team}_mid_xp'] = (total_xp / game_time) * mid_weight * 1.2

        # Hero Efficiency Metrics
        for pos in range(1,6):
            # Farm efficiency with position weight
            pos_weight = 1.2 if pos <= 3 else 0.8
            features[f'{team}{pos}_farm_eff'] = (
                df[f'{team}{pos}_gold'] * 0.7 +
                df[f'{team}{pos}_lh'] * 0.3
            ) / game_time * pos_weight

            # Hero win rate features
            features[f'{team}{pos}_hero_win_rate'] = df[f'{team}{pos}_hero_id'].map(h
            features[f'{team}{pos}_hero_id'] = df[f'{team}{pos}_hero_id']

        # Team Composition Strength
        features[f'{team}_avg_win_rate'] = np.mean([
            features[f'{team}{pos}_hero_win_rate'] for pos in range(1,6)
        ], axis=0)

        # Objective Control
        features[f'{team}_objective_score'] = (
            df[[f'{team}{i}_towers_killed' for i in range(1,6)]].sum(axis=1) * 0.6 +
            df[[f'{team}{i}_roshans_killed' for i in range(1,6)]].sum(axis=1) * 0.4
        )

    # Advantage Calculations
    for metric in ['core_early_gold', 'core_early_xp', 'support_early_gold',
                   'support_early_xp', 'mid_gold', 'mid_xp', 'avg_win_rate']:
        features[f'{metric}_diff'] = features[f'r_{metric}'] - features[f'd_{metric}'
        features[f'{metric}_ratio'] = (features[f'r_{metric}'] + 1e-6) / (features[f'

    # Game State Features
    features['game_time'] = game_time
    features['game_time_sq'] = game_time ** 2
    features['game_mode'] = df['game_mode']

    return pd.DataFrame(features)

# Prepare data
X_train = create_enhanced_features(train_df)
X_test = create_enhanced_features(test_df)
y_train = train_df['radiant_win']
```

In [27]:
```
#Define Categorical Features
cat_features = [col for col in X_train.columns if 'hero_id' in col or 'game_mode' in
```

In [28]:
```
#Define Objective Function for Optuna
def objective(trial):
    params = {
        'iterations': trial.suggest_int('iterations', 1000, 10000),
        'learning_rate': trial.suggest_loguniform('learning_rate', 0.001, 0.1),
        'depth': trial.suggest_int('depth', 4, 10),
        'l2_leaf_reg': trial.suggest_loguniform('l2_leaf_reg', 1, 10),
        'min_data_in_leaf': trial.suggest_int('min_data_in_leaf', 20, 100),
        'random_strength': trial.suggest_uniform('random_strength', 0.1, 1.0),
        'bagging_temperature': trial.suggest_uniform('bagging_temperature', 0.1, 1.0)
        'grow_policy': trial.suggest_categorical('grow_policy', ['SymmetricTree', 'De
        'random_seed': 42,
        'verbose': False
    }
    model = CatBoostClassifier(**params)

    X_train_split, X_val_split, y_train_split, y_val_split = train_test_split(X_train

    model.fit(
        X_train_split,
        y_train_split,
        eval_set=(X_val_split, y_val_split),
        early_stopping_rounds=200,
        cat_features=cat_features,
        verbose=False
    )

    val_pred = model.predict_proba(X_val_split)[:, 1]
    auc = roc_auc_score(y_val_split, val_pred)

    return auc
```

In [29]:
```
#Run Optuna Optimization
study = optuna.create_study(direction='maximize')
study.optimize(objective, n_trials=50)

# Get the best parameters
best_params = study.best_params
```

```
[I 2025-02-12 19:50:30,198] A new study created in memory with name: no-name-8784bb88-f075-416d-9d8d-a9c7063aeb0e
[I 2025-02-12 19:50:44,461] Trial 0 finished with value: 0.840116819500469 and parameters: {'iterations': 6108, 'learning_rate': 0.016136899498313415, 'depth': 4, 'l2_leaf_reg': 8.446318308362173, 'min_data_in_leaf': 22, 'random_strength': 0.8666159283141688, 'bagging_temperature': 0.6906527112875646, 'grow_policy': 'Lossguide'}. Best is trial 0 with value: 0.840116819500469.
[I 2025-02-12 19:51:04,311] Trial 1 finished with value: 0.8391678231883681 and parameters: {'iterations': 8581, 'learning_rate': 0.0070169087128931435, 'depth': 6, 'l2_leaf_reg': 1.4622884021659195, 'min_data_in_leaf': 51, 'random_strength': 0.186827261534519, 'bagging_temperature': 0.5330639518753737, 'grow_policy': 'Lossguide'}. Best is trial 0 with value: 0.840116819500469.
[I 2025-02-12 19:55:58,528] Trial 2 finished with value: 0.838903865456193 and parameters: {'iterations': 5951, 'learning_rate': 0.0010325389691269178, 'depth': 10, 'l2_leaf_reg': 1.0235172131803476, 'min_data_in_leaf': 27, 'random_strength': 0.14672346173641235, 'bagging_temperature': 0.11257164111035663, 'grow_policy': 'SymmetricTree'}. Best is trial 0 with value: 0.840116819500469.
[I 2025-02-12 19:56:01,823] Trial 3 finished with value: 0.8382963075726517 and parameters: {'iterations': 3835, 'learning_rate': 0.06024574635936515, 'depth': 5, 'l2_leaf_reg': 2.643255667939475, 'min_data_in_leaf': 24, 'random_strength': 0.3793038468797767, 'bagging_temperature': 0.561516469110466, 'grow_policy': 'Depthwise'}. Best is trial 0 with value: 0.840116819500469.
[I 2025-02-12 19:59:31,531] Trial 4 finished with value: 0.8395877662872729 and parameters: {'iterations': 5054, 'learning_rate': 0.0014040764357510708, 'depth': 8, 'l2_leaf_reg': 7.263857393006822, 'min_data_in_leaf': 21, 'random_strength': 0.3512751085266774, 'bagging_temperature': 0.8172135363589489, 'grow_policy': 'Lossguide'}. Best is trial 0 with value: 0.840116819500469.
[I 2025-02-12 20:01:40,280] Trial 5 finished with value: 0.8393631746651549 and parameters: {'iterations': 5025, 'learning_rate': 0.0014440363885708944, 'depth': 8, 'l2_leaf_reg': 2.691391948291677, 'min_data_in_leaf': 31, 'random_strength': 0.11173220196351924, 'bagging_temperature': 0.34856862638493513, 'grow_policy': 'SymmetricTree'}. Best is trial 0 with value: 0.840116819500469.
[I 2025-02-12 20:02:09,375] Trial 6 finished with value: 0.8397786805435616 and parameters: {'iterations': 9841, 'learning_rate': 0.01532091517656213, 'depth': 8, 'l2_leaf_reg': 1.8090475798680163, 'min_data_in_leaf': 86, 'random_strength': 0.9520780655340746, 'bagging_temperature': 0.45197168483650363, 'grow_policy': 'SymmetricTree'}. Best is trial 0 with value: 0.840116819500469.
[I 2025-02-12 20:02:30,533] Trial 7 finished with value: 0.8402691640708927 and parameters: {'iterations': 5638, 'learning_rate': 0.01576771117496574, 'depth': 4, 'l2_leaf_reg': 4.276858086261457, 'min_data_in_leaf': 80, 'random_strength': 0.841672733386203, 'bagging_temperature': 0.22268708821269015, 'grow_policy': 'SymmetricTree'}. Best is trial 7 with value: 0.8402691640708927.
[I 2025-02-12 20:03:55,070] Trial 8 finished with value: 0.8392202734101495 and parameters: {'iterations': 6401, 'learning_rate': 0.007106411308149588, 'depth': 10, 'l2_leaf_reg': 4.1161448857208525, 'min_data_in_leaf': 97, 'random_strength': 0.12741946929412861, 'bagging_temperature': 0.4420618145588583, 'grow_policy': 'SymmetricTree'}. Best is trial 7 with value: 0.8402691640708927.
[I 2025-02-12 20:04:06,330] Trial 9 finished with value: 0.8369534226102097 and parameters: {'iterations': 1031, 'learning_rate': 0.08561619794116097, 'depth': 9, 'l2_leaf_reg': 4.677860954568512, 'min_data_in_leaf': 21, 'random_strength': 0.27766251458879343, 'bagging_temperature': 0.6277468871047887, 'grow_policy': 'SymmetricTree'}. Best is trial 7 with value: 0.8402691640708927.
[I 2025-02-12 20:04:12,072] Trial 10 finished with value: 0.8397828902142905 and parameters: {'iterations': 2388, 'learning_rate': 0.03302969810959035, 'depth': 4, 'l2_leaf_reg': 5.628534155558901, 'min_data_in_leaf': 72, 'random_strength': 0.7177954147034526, 'bagging_temperature': 0.14692506863080623, 'grow_policy': 'Depthwise'}. Best is trial 7 with value: 0.8402691640708927.
[I 2025-02-12 20:04:27,704] Trial 11 finished with value: 0.8403758849126084 and parameters: {'iterations': 7584, 'learning_rate': 0.01784220258669064, 'depth': 4, 'l2_leaf_reg': 9.84342974139258, 'min_data_in_leaf': 55, 'random_strength': 0.8997196864300747, 'bagging_temperature': 0.9425535690730353, 'grow_policy': 'Lossguide'}. Best is trial 11 with value: 0.8403758849126084.
[I 2025-02-12 20:05:18,565] Trial 12 finished with value: 0.8399693672500815 and parameters: {'iterations': 7718, 'learning_rate': 0.0036131760581170416, 'depth': 6, 'l2_leaf_reg': 8.9993262299375, 'min_data_in_leaf': 57, 'random_strength': 0.753987043402723
```

8, 'bagging_temperature': 0.9466880229862766, 'grow_policy': 'Lossguide'}. Best is tri
al 11 with value: 0.8403758849126084.
[I 2025-02-12 20:05:26,137] Trial 13 finished with value: 0.8391564456999123 and param
eters: {'iterations': 7504, 'learning_rate': 0.03167049504096631, 'depth': 5, 'l2_leaf
_reg': 3.48949600862125, 'min_data_in_leaf': 70, 'random_strength': 0.621381490567442
4, 'bagging_temperature': 0.27777239327178194, 'grow_policy': 'Lossguide'}. Best is tr
ial 11 with value: 0.8403758849126084.
[I 2025-02-12 20:05:31,200] Trial 14 finished with value: 0.8399768763924623 and param
eters: {'iterations': 3959, 'learning_rate': 0.02720874318578251, 'depth': 4, 'l2_leaf
_reg': 6.719686285227397, 'min_data_in_leaf': 44, 'random_strength': 0.96621626374357
6, 'bagging_temperature': 0.9775810301143932, 'grow_policy': 'Depthwise'}. Best is tri
al 11 with value: 0.8403758849126084.
[I 2025-02-12 20:07:07,781] Trial 15 finished with value: 0.8402941945454956 and param
eters: {'iterations': 7546, 'learning_rate': 0.0036414548805247396, 'depth': 6, 'l2_le
af_reg': 9.781470194988852, 'min_data_in_leaf': 74, 'random_strength': 0.5335892429400
164, 'bagging_temperature': 0.7760087256233625, 'grow_policy': 'SymmetricTree'}. Best
is trial 11 with value: 0.8403758849126084.
[I 2025-02-12 20:08:06,574] Trial 16 finished with value: 0.8401566407100645 and param
eters: {'iterations': 9464, 'learning_rate': 0.0033341543115904307, 'depth': 6, 'l2_le
af_reg': 5.823472631406535, 'min_data_in_leaf': 41, 'random_strength': 0.4883384202844
493, 'bagging_temperature': 0.8031427717828161, 'grow_policy': 'Lossguide'}. Best is t
rial 11 with value: 0.8403758849126084.
[I 2025-02-12 20:09:49,232] Trial 17 finished with value: 0.8401403709015728 and param
eters: {'iterations': 7352, 'learning_rate': 0.003085713993094283, 'depth': 7, 'l2_lea
f_reg': 9.730120718149262, 'min_data_in_leaf': 65, 'random_strength': 0.54503686179234
35, 'bagging_temperature': 0.8321743982644865, 'grow_policy': 'SymmetricTree'}. Best i
s trial 11 with value: 0.8403758849126084.
[I 2025-02-12 20:10:26,113] Trial 18 finished with value: 0.840263930426203 and parame
ters: {'iterations': 8528, 'learning_rate': 0.005924990898438478, 'depth': 5, 'l2_leaf
_reg': 7.460349197299446, 'min_data_in_leaf': 82, 'random_strength': 0.677910864257727
5, 'bagging_temperature': 0.7106932719772387, 'grow_policy': 'Lossguide'}. Best is tri
al 11 with value: 0.8403758849126084.
[I 2025-02-12 20:10:44,543] Trial 19 finished with value: 0.8396343002150572 and param
eters: {'iterations': 6918, 'learning_rate': 0.009989465005642108, 'depth': 7, 'l2_lea
f_reg': 9.99981249861367, 'min_data_in_leaf': 59, 'random_strength': 0.51023058939527
3, 'bagging_temperature': 0.9050795037636967, 'grow_policy': 'Depthwise'}. Best is tri
al 11 with value: 0.8403758849126084.
[I 2025-02-12 20:11:57,372] Trial 20 finished with value: 0.8399232884218355 and param
eters: {'iterations': 8579, 'learning_rate': 0.0021046428439238985, 'depth': 5, 'l2_le
af_reg': 5.701078434003654, 'min_data_in_leaf': 100, 'random_strength': 0.447263624037
9829, 'bagging_temperature': 0.7401224824269813, 'grow_policy': 'Lossguide'}. Best is
trial 11 with value: 0.8403758849126084.
[I 2025-02-12 20:12:12,731] Trial 21 finished with value: 0.8401911145000858 and param
eters: {'iterations': 4104, 'learning_rate': 0.01805685814670291, 'depth': 4, 'l2_leaf
_reg': 4.5037419233569285, 'min_data_in_leaf': 82, 'random_strength': 0.81721892981506
12, 'bagging_temperature': 0.897856410430868, 'grow_policy': 'SymmetricTree'}. Best is
trial 11 with value: 0.8403758849126084.
[I 2025-02-12 20:12:42,216] Trial 22 finished with value: 0.8402800864598101 and param
eters: {'iterations': 8057, 'learning_rate': 0.010451184725495558, 'depth': 5, 'l2_lea
f_reg': 1.9787620996790816, 'min_data_in_leaf': 74, 'random_strength': 0.8748289159652
083, 'bagging_temperature': 0.2194597174264517, 'grow_policy': 'SymmetricTree'}. Best
is trial 11 with value: 0.8403758849126084.
[I 2025-02-12 20:13:32,972] Trial 23 finished with value: 0.840258014132206 and parame
ters: {'iterations': 8229, 'learning_rate': 0.004935128849110394, 'depth': 6, 'l2_leaf
_reg': 1.909008723796015, 'min_data_in_leaf': 73, 'random_strength': 0.638185864403550
1, 'bagging_temperature': 0.9905094994818276, 'grow_policy': 'SymmetricTree'}. Best is
trial 11 with value: 0.8403758849126084.
[I 2025-02-12 20:14:01,372] Trial 24 finished with value: 0.8405149178215386 and param
eters: {'iterations': 9160, 'learning_rate': 0.010270110683315797, 'depth': 5, 'l2_lea
f_reg': 2.156218322064168, 'min_data_in_leaf': 92, 'random_strength': 0.88083353952539
81, 'bagging_temperature': 0.8881353073486111, 'grow_policy': 'SymmetricTree'}. Best i
s trial 24 with value: 0.8405149178215386.
[I 2025-02-12 20:14:38,486] Trial 25 finished with value: 0.8400896273030598 and param
eters: {'iterations': 9368, 'learning_rate': 0.0103371804635015, 'depth': 6, 'l2_leaf_
reg': 2.4276511856981817, 'min_data_in_leaf': 93, 'random_strength': 0.994511418704884
7, 'bagging_temperature': 0.8579913168153553, 'grow_policy': 'SymmetricTree'}. Best is

```
trial 24 with value: 0.8405149178215386.
[I 2025-02-12 20:14:55,300] Trial 26 finished with value: 0.840242540747906 and parame
ters: {'iterations': 6865, 'learning_rate': 0.023164423353495465, 'depth': 5, 'l2_leaf
_reg': 1.439812739585279, 'min_data_in_leaf': 92, 'random_strength': 0.595736980109542
1, 'bagging_temperature': 0.7658082542038224, 'grow_policy': 'SymmetricTree'}. Best is
trial 24 with value: 0.8405149178215386.
[I 2025-02-12 20:15:07,339] Trial 27 finished with value: 0.8396038085459956 and param
eters: {'iterations': 9232, 'learning_rate': 0.039778352553750604, 'depth': 7, 'l2_lea
f_reg': 3.585898511140938, 'min_data_in_leaf': 65, 'random_strength': 0.772825831232561
1, 'bagging_temperature': 0.6511436007191285, 'grow_policy': 'SymmetricTree'}. Best i
s trial 24 with value: 0.8405149178215386.
[I 2025-02-12 20:15:44,140] Trial 28 finished with value: 0.840312853626563 and parame
ters: {'iterations': 9269, 'learning_rate': 0.004741802703234369, 'depth': 5, 'l2_leaf
_reg': 2.9793547059230647, 'min_data_in_leaf': 51, 'random_strength': 0.909916125345022
9, 'bagging_temperature': 0.9160750216237548, 'grow_policy': 'Depthwise'}. Best is tr
ial 24 with value: 0.8405149178215386.
[I 2025-02-12 20:15:57,138] Trial 29 finished with value: 0.8397549015926891 and param
eters: {'iterations': 9927, 'learning_rate': 0.012719804828744479, 'depth': 4, 'l2_lea
f_reg': 2.9501145609990655, 'min_data_in_leaf': 52, 'random_strength': 0.8975734611132
241, 'bagging_temperature': 0.9034855915614565, 'grow_policy': 'Depthwise'}. Best is t
rial 24 with value: 0.8405149178215386.
[I 2025-02-12 20:16:09,369] Trial 30 finished with value: 0.8402581279070904 and param
eters: {'iterations': 8908, 'learning_rate': 0.020995118986986222, 'depth': 4, 'l2_lea
f_reg': 2.377708818618802, 'min_data_in_leaf': 40, 'random_strength': 0.914291026538443
4, 'bagging_temperature': 0.9961259583270871, 'grow_policy': 'Depthwise'}. Best is tr
ial 24 with value: 0.8405149178215386.
[I 2025-02-12 20:16:43,334] Trial 31 finished with value: 0.840328327010863 and parame
ters: {'iterations': 6999, 'learning_rate': 0.004491127513762571, 'depth': 5, 'l2_leaf
_reg': 8.265034524008277, 'min_data_in_leaf': 50, 'random_strength': 0.815102026438289
1, 'bagging_temperature': 0.9189719837974684, 'grow_policy': 'Depthwise'}. Best is tri
al 24 with value: 0.8405149178215386.
[I 2025-02-12 20:17:06,953] Trial 32 finished with value: 0.8405179897434216 and param
eters: {'iterations': 6683, 'learning_rate': 0.0076517313589637365, 'depth': 5, 'l2_le
af_reg': 7.703144559140397, 'min_data_in_leaf': 48, 'random_strength': 0.773719660173077
73, 'bagging_temperature': 0.8747240514805957, 'grow_policy': 'Depthwise'}. Best is t
rial 32 with value: 0.8405179897434216.
[I 2025-02-12 20:17:32,979] Trial 33 finished with value: 0.840226043389645 and parame
ters: {'iterations': 6432, 'learning_rate': 0.007073386615817629, 'depth': 5, 'l2_leaf
_reg': 8.206620485948218, 'min_data_in_leaf': 46, 'random_strength': 0.805147251989880
5, 'bagging_temperature': 0.8611268165821716, 'grow_policy': 'Depthwise'}. Best is tri
al 32 with value: 0.8405179897434216.
[I 2025-02-12 20:17:50,030] Trial 34 finished with value: 0.8397795907426383 and param
eters: {'iterations': 6922, 'learning_rate': 0.00839308082381201, 'depth': 4, 'l2_leaf
_reg': 8.169691149730921, 'min_data_in_leaf': 34, 'random_strength': 0.724528343636517
3, 'bagging_temperature': 0.945042952854767, 'grow_policy': 'Depthwise'}. Best is tria
l 32 with value: 0.8405179897434216.
[I 2025-02-12 20:18:15,379] Trial 35 finished with value: 0.8401081726092425 and param
eters: {'iterations': 6054, 'learning_rate': 0.005305396067026149, 'depth': 5, 'l2_lea
f_reg': 1.095481759492582, 'min_data_in_leaf': 56, 'random_strength': 0.834715302312615
5, 'bagging_temperature': 0.6851321110488148, 'grow_policy': 'Depthwise'}. Best is tr
ial 32 with value: 0.8405179897434216.
[I 2025-02-12 20:18:32,085] Trial 36 finished with value: 0.8401215980456206 and param
eters: {'iterations': 7928, 'learning_rate': 0.012591341330481155, 'depth': 6, 'l2_lea
f_reg': 6.735289203150057, 'min_data_in_leaf': 64, 'random_strength': 0.776609591413304
5, 'bagging_temperature': 0.861226323502286, 'grow_policy': 'Lossguide'}. Best is tri
al 32 with value: 0.8405179897434216.
[I 2025-02-12 20:19:08,743] Trial 37 finished with value: 0.8399490015457456 and param
eters: {'iterations': 4868, 'learning_rate': 0.0024800892303757016, 'depth': 4, 'l2_le
af_reg': 1.5345902718040434, 'min_data_in_leaf': 48, 'random_strength': 0.945381813598
197, 'bagging_temperature': 0.6101103205568552, 'grow_policy': 'Depthwise'}. Best is t
rial 32 with value: 0.8405179897434216.
[I 2025-02-12 20:19:15,265] Trial 38 finished with value: 0.8393292697495565 and param
eters: {'iterations': 5538, 'learning_rate': 0.05101563196440233, 'depth': 6, 'l2_leaf
_reg': 5.135668123447489, 'min_data_in_leaf': 39, 'random_strength': 0.677237887122313
5, 'bagging_temperature': 0.5353201015632082, 'grow_policy': 'Lossguide'}. Best is tri
al 32 with value: 0.8405179897434216.
```

```
[I 2025-02-12 20:20:18,307] Trial 39 finished with value: 0.840239696375792 and parame
ters: {'iterations': 6615, 'learning_rate': 0.0018088001000109352, 'depth': 5, 'l2_lea
f_reg': 7.500338296234908, 'min_data_in_leaf': 35, 'random_strength': 0.87266687577873
66, 'bagging_temperature': 0.804800017489462, 'grow_policy': 'Depthwise'}. Best is tri
al 32 with value: 0.8405179897434216.
[I 2025-02-12 20:20:39,288] Trial 40 finished with value: 0.8401457183211468 and param
eters: {'iterations': 7322, 'learning_rate': 0.008160642984981068, 'depth': 4, 'l2_lea
f_reg': 6.4347012080855, 'min_data_in_leaf': 55, 'random_strength': 0.994486910468515
8, 'bagging_temperature': 0.45786364357642956, 'grow_policy': 'Depthwise'}. Best is tr
ial 32 with value: 0.8405179897434216.
[I 2025-02-12 20:21:16,925] Trial 41 finished with value: 0.840160622831024 and parame
ters: {'iterations': 8941, 'learning_rate': 0.004544215343728066, 'depth': 5, 'l2_leaf
_reg': 2.3238147984103152, 'min_data_in_leaf': 51, 'random_strength': 0.92001466089156
86, 'bagging_temperature': 0.9385870913509222, 'grow_policy': 'Depthwise'}. Best is tr
ial 32 with value: 0.8405179897434216.
[I 2025-02-12 20:21:52,636] Trial 42 finished with value: 0.8402812242086558 and param
eters: {'iterations': 8202, 'learning_rate': 0.004293562354736035, 'depth': 5, 'l2_lea
f_reg': 3.0982394849336874, 'min_data_in_leaf': 49, 'random_strength': 0.8578982625845
509, 'bagging_temperature': 0.898450105122085, 'grow_policy': 'Depthwise'}. Best is tr
ial 32 with value: 0.8405179897434216.
[I 2025-02-12 20:22:30,626] Trial 43 finished with value: 0.8396456777035131 and param
eters: {'iterations': 5868, 'learning_rate': 0.005891483970324492, 'depth': 9, 'l2_lea
f_reg': 3.799049019655831, 'min_data_in_leaf': 61, 'random_strength': 0.79759191988975
04, 'bagging_temperature': 0.945066289802122, 'grow_policy': 'Depthwise'}. Best is tri
al 32 with value: 0.8405179897434216.
[I 2025-02-12 20:22:45,803] Trial 44 finished with value: 0.8400666447763789 and param
eters: {'iterations': 8994, 'learning_rate': 0.014278129241013316, 'depth': 5, 'l2_lea
f_reg': 2.7524826388084187, 'min_data_in_leaf': 53, 'random_strength': 0.7332238269910
353, 'bagging_temperature': 0.8588945207624913, 'grow_policy': 'Depthwise'}. Best is t
rial 32 with value: 0.8405179897434216.
[I 2025-02-12 20:23:48,251] Trial 45 finished with value: 0.8402437922716361 and param
eters: {'iterations': 9601, 'learning_rate': 0.0026567971597554283, 'depth': 4, 'l2_le
af_reg': 8.758902347239852, 'min_data_in_leaf': 44, 'random_strength': 0.9425331661703
582, 'bagging_temperature': 0.9213702918398171, 'grow_policy': 'Lossguide'}. Best is t
rial 32 with value: 0.8405179897434216.
[I 2025-02-12 20:24:04,659] Trial 46 finished with value: 0.8401489040179144 and param
eters: {'iterations': 5239, 'learning_rate': 0.008878473023351633, 'depth': 5, 'l2_lea
f_reg': 2.0478981910969103, 'min_data_in_leaf': 59, 'random_strength': 0.8408596017345
434, 'bagging_temperature': 0.8210202402382967, 'grow_policy': 'Depthwise'}. Best is t
rial 32 with value: 0.8405179897434216.
[I 2025-02-12 20:24:18,103] Trial 47 finished with value: 0.8399621994323544 and param
eters: {'iterations': 8600, 'learning_rate': 0.017976252709374654, 'depth': 6, 'l2_lea
f_reg': 1.615162234864446, 'min_data_in_leaf': 69, 'random_strength': 0.89301014851632
73, 'bagging_temperature': 0.748303635980065, 'grow_policy': 'Lossguide'}. Best is tri
al 32 with value: 0.8405179897434216.
[I 2025-02-12 20:25:00,526] Trial 48 finished with value: 0.8400376321808165 and param
eters: {'iterations': 7253, 'learning_rate': 0.00410967601698029, 'depth': 4, 'l2_leaf
_reg': 3.334035198549425, 'min_data_in_leaf': 43, 'random_strength': 0.95818096739491
4, 'bagging_temperature': 0.9543447191307586, 'grow_policy': 'Depthwise'}. Best is tri
al 32 with value: 0.8405179897434216.
[I 2025-02-12 20:25:18,025] Trial 49 finished with value: 0.8394430446341147 and param
eters: {'iterations': 4541, 'learning_rate': 0.007284043974688932, 'depth': 4, 'l2_lea
f_reg': 7.910687688174968, 'min_data_in_leaf': 26, 'random_strength': 0.20518735359589
313, 'bagging_temperature': 0.9957106705953265, 'grow_policy': 'Depthwise'}. Best is t
rial 32 with value: 0.8405179897434216.
```

In [30]:

```python
#Training Setup and Model Training
n_folds = 7
predictions = np.zeros(len(X_test))
oof_predictions = np.zeros(len(X_train))
skf = StratifiedKFold(n_splits=n_folds, shuffle=True, random_state=42)

for fold, (train_idx, val_idx) in enumerate(skf.split(X_train, y_train)):
    print(f"\nFold {fold + 1}:")

    X_fold_train = X_train.iloc[train_idx]
```

```
    y_fold_train = y_train.iloc[train_idx]
    X_fold_val = X_train.iloc[val_idx]
    y_fold_val = y_train.iloc[val_idx]

    model = CatBoostClassifier(**best_params)

    model.fit(
        X_fold_train,
        y_fold_train,
        eval_set=(X_fold_val, y_fold_val),
        early_stopping_rounds=200,
        cat_features=cat_features,
        verbose=False
    )

    val_pred = model.predict_proba(X_fold_val)[:, 1]
    oof_predictions[val_idx] = val_pred
    print(f"Validation AUC: {roc_auc_score(y_fold_val, val_pred):.4f}")

    predictions += model.predict_proba(X_test)[:, 1] / n_folds

print(f"\nOverall CV score: {roc_auc_score(y_train, oof_predictions):.4f}")
```

```
Fold 1:
Validation AUC: 0.8436

Fold 2:
Validation AUC: 0.8381

Fold 3:
Validation AUC: 0.8494

Fold 4:
Validation AUC: 0.8414

Fold 5:
Validation AUC: 0.8412

Fold 6:
Validation AUC: 0.8464

Fold 7:
Validation AUC: 0.8300

Overall CV score: 0.8415
```

In [31]:
```
# Create submission
submission = pd.DataFrame({
    'ID': test_df['ID'],
    'radiant_win': predictions
})
submission.to_csv('dota2_kaggle.csv', index=False)
print("\nOptimized submission file created!")
```

```
Optimized submission file created!
```

In [3]:
```
!jupyter nbconvert --to html Dota2_Kaggle.ipynb
```

```
Traceback (most recent call last):
  File "/Users/aryaman/Library/Python/3.9/bin/jupyter-nbconvert", line 5, in <module>
    from nbconvert.nbconvertapp import main
  File "/Users/aryaman/Library/Python/3.9/lib/python/site-packages/nbconvert/nbconvert
app.py", line 193, in <module>
    class NbConvertApp(JupyterApp):
  File "/Users/aryaman/Library/Python/3.9/lib/python/site-packages/nbconvert/nbconvert
app.py", line 252, in NbConvertApp
    Options include {get_export_names()}.
  File "/Users/aryaman/Library/Python/3.9/lib/python/site-packages/nbconvert/exporter
s/base.py", line 145, in get_export_names
    e = get_exporter(exporter_name)(config=config)
  File "/Users/aryaman/Library/Python/3.9/lib/python/site-packages/nbconvert/exporter
s/base.py", line 106, in get_exporter
    exporter = items[0].load()
  File "/Users/aryaman/Library/Python/3.9/lib/python/site-packages/importlib_metadata/
__init__.py", line 189, in load
    module = import_module(match.group('module'))
  File "/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versi
ons/3.9/lib/python3.9/importlib/__init__.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "/Users/aryaman/Library/Python/3.9/lib/python/site-packages/jupyter_contrib_nbe
xtensions/nbconvert_support/__init__.py", line 5, in <module>
    from .collapsible_headings import ExporterCollapsibleHeadings
  File "/Users/aryaman/Library/Python/3.9/lib/python/site-packages/jupyter_contrib_nbe
xtensions/nbconvert_support/collapsible_headings.py", line 6, in <module>
    from notebook.services.config import ConfigManager
ModuleNotFoundError: No module named 'notebook.services'
```