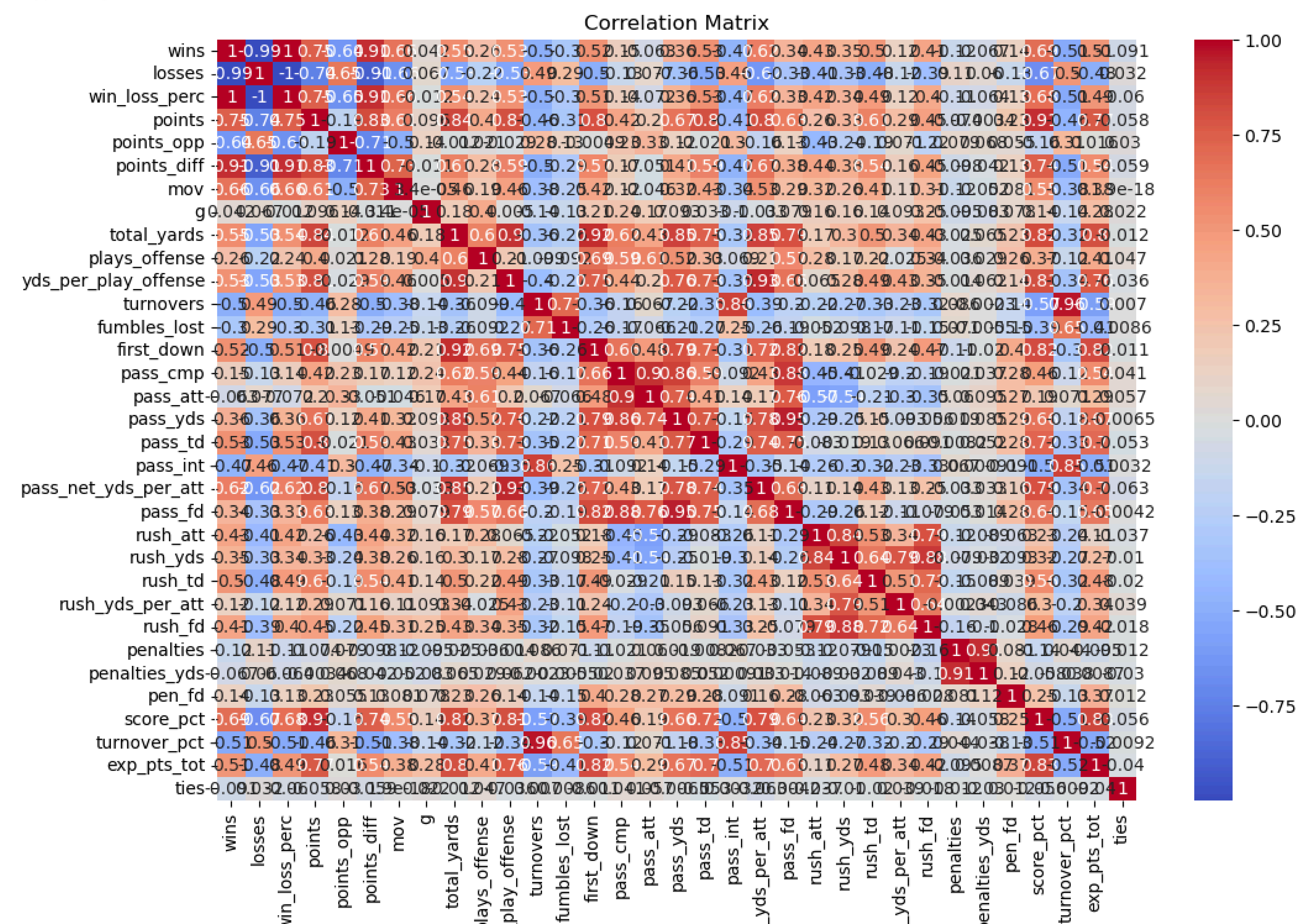
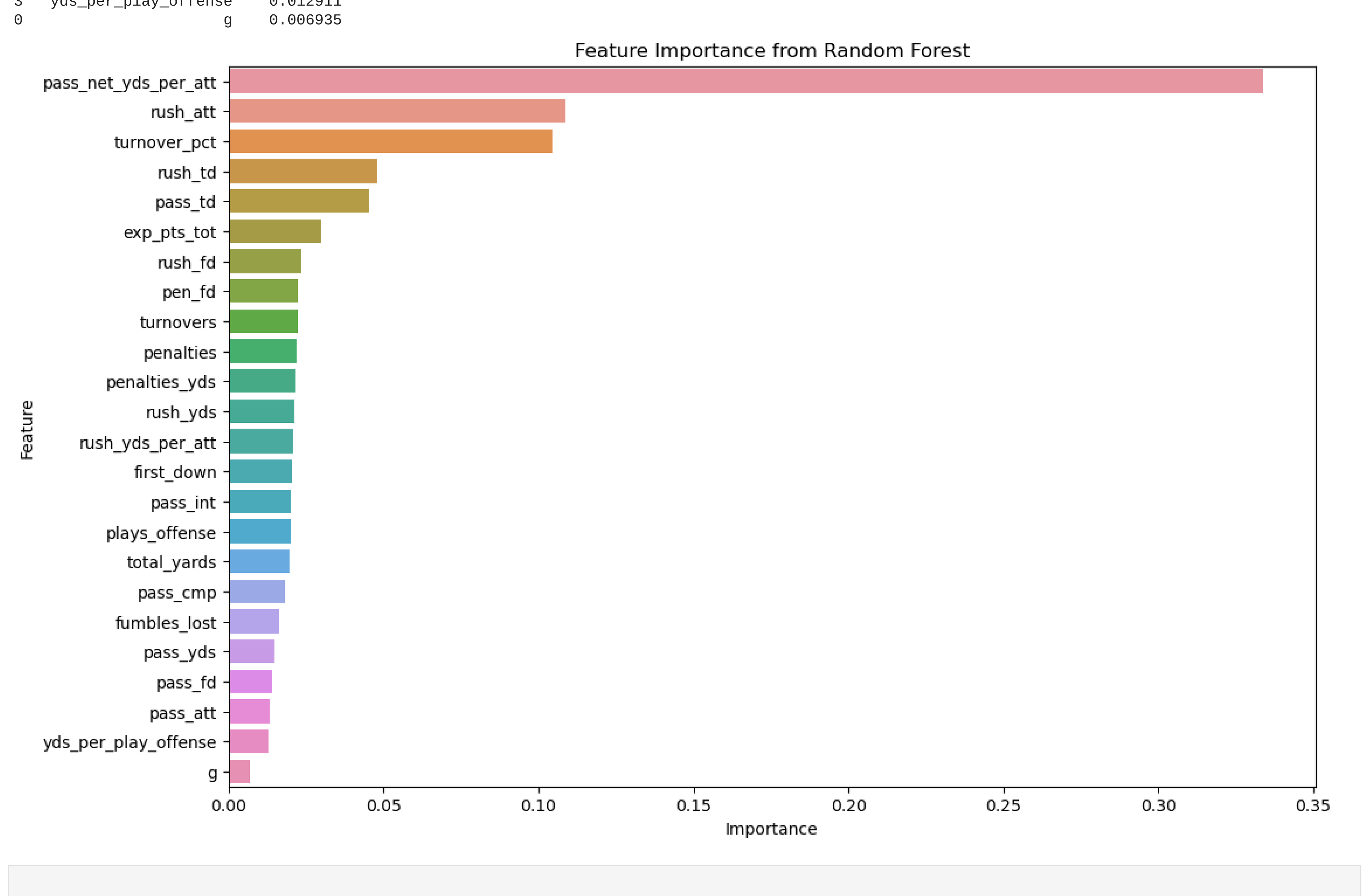
In [5]:

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.feature_selection import SelectKBest, f_regression

# Load the dataset
df = pd.read_csv('team_stats_2003_2023.csv')

# Display the first few rows of the dataset and column names
print(df.head())
print(df.columns)

# Check for missing values
print(df.isnull().sum())

# Inspect the column names to identify non-numeric columns
print(df.columns)

# Ensure column names are correctly identified
columns_to_drop = ['team']
for col in df.columns:
    if 'year' in col.lower():
        columns_to_drop.append(col)

print(f"Columns to drop: {columns_to_drop}")

# Drop non-numeric columns if they exist (adjust as necessary based on actual column names)
df = df.drop(columns=columns_to_drop)

# If there are missing values, handle them (e.g., fill with mean, median, or drop)
df = df.fillna(df.mean())

# Verify all columns are numeric now
print(df.dtypes)

# Define the features (X) and the target variable (y)
X = df.drop(columns=['win_loss_perc', 'wins', 'losses', 'ties', 'points_diff', 'points', 'points_opp', 'mov', 'score_pct'])  # Adjust based on your
y = df['win_loss_perc']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Normalize/scale the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Feature selection using correlation
correlation_matrix = df.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

# Feature selection using SelectKBest
selector = SelectKBest(score_func=f_regression, k='all')
selector.fit(X_train_scaled, y_train)
feature_scores = pd.DataFrame({'Feature': X.columns, 'Score': selector.scores_})
feature_scores = feature_scores.sort_values(by='Score', ascending=False)
print(feature_scores)

# Fit a linear regression model
lr = LinearRegression()
lr.fit(X_train_scaled, y_train)
y_pred_lr = lr.predict(X_test_scaled)

# Evaluate the linear regression model
print('Linear Regression RMSE:', np.sqrt(mean_squared_error(y_test, y_pred_lr)))
print('Linear Regression R^2:', r2_score(y_test, y_pred_lr))

# Fit a random forest regressor model
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train_scaled, y_train)
y_pred_rf = rf.predict(X_test_scaled)

# Evaluate the random forest model
print('Random Forest RMSE:', np.sqrt(mean_squared_error(y_test, y_pred_rf)))
print('Random Forest R^2:', r2_score(y_test, y_pred_rf))

# Feature importance from random forest
feature_importances = pd.DataFrame({'Feature': X.columns, 'Importance': rf.feature_importances_})
feature_importances = feature_importances.sort_values(by='Importance', ascending=False)
print(feature_importances)

# Plot feature importance
plt.figure(figsize=(12, 8))
sns.barplot(x='Importance', y='Feature', data=feature_importances)
plt.title('Feature Importance from Random Forest')
plt.show()
```

```
   year                  team  wins  losses  win_loss_perc  points  \
0  2003  New England Patriots    14       2          0.875     348
1  2003        Miami Dolphins    10       6          0.625     311
2  2003         Buffalo Bills     6      10          0.375     243
3  2003         New York Jets     6      10          0.375     283
4  2003      Baltimore Ravens    10       6          0.625     391

   points_opp  points_diff  mov   g  ...  rush_td  rush_yds_per_att  rush_fd  \
0         238          110  6.9  16  ...        9               3.4       91
1         261           50  3.1  16  ...       14               3.7       99
2         279          -36 -2.3  16  ...       13               3.9       96
3         299          -16 -1.0  16  ...        8               4.0       78
4         281          110  6.9  16  ...       18               4.8      115

   penalties  penalties_yds  pen_fd  score_pct  turnover_pct  exp_pts_tot  \
0        111            998      26       27.9          11.3      -136.51
1        103            913      22       28.1          17.2      -177.92
2        106            891      22       21.9          17.6      -230.07
3         69            550      15       32.4          11.8      -107.89
4        126            970      23       31.8          16.6      -220.50

   ties
0   NaN
1   NaN
2   NaN
3   NaN
4   NaN

[5 rows x 35 columns]
Index(['year', 'team', 'wins', 'losses', 'win_loss_perc', 'points',
       'points_opp', 'points_diff', 'mov', 'g', 'total_yards', 'plays_offense',
       'yds_per_play_offense', 'turnovers', 'fumbles_lost', 'first_down',
       'pass_cmp', 'pass_att', 'pass_yds', 'pass_td', 'pass_int',
       'pass_net_yds_per_att', 'pass_fd', 'rush_att', 'rush_yds', 'rush_td',
       'rush_yds_per_att', 'rush_fd', 'penalties', 'penalties_yds', 'pen_fd',
       'score_pct', 'turnover_pct', 'exp_pts_tot', 'ties'],
      dtype='object')
year                      0
team                      0
wins                      0
losses                    0
win_loss_perc             0
points                    0
points_opp                0
points_diff               0
mov                     320
g                         0
total_yards               0
plays_offense             0
yds_per_play_offense      0
turnovers                 0
fumbles_lost              0
first_down                0
pass_cmp                  0
pass_att                  0
pass_yds                  0
pass_td                   0
pass_int                  0
pass_net_yds_per_att      0
pass_fd                   0
rush_att                  0
rush_yds                  0
rush_td                   0
rush_yds_per_att          0
rush_fd                   0
penalties                 0
penalties_yds             0
pen_fd                    0
score_pct                 0
turnover_pct              0
exp_pts_tot               0
ties                    352
dtype: int64
Index(['year', 'team', 'wins', 'losses', 'win_loss_perc', 'points',
       'points_opp', 'points_diff', 'mov', 'g', 'total_yards', 'plays_offense',
       'yds_per_play_offense', 'turnovers', 'fumbles_lost', 'first_down',
       'pass_cmp', 'pass_att', 'pass_yds', 'pass_td', 'pass_int',
       'pass_net_yds_per_att', 'pass_fd', 'rush_att', 'rush_yds', 'rush_td',
       'rush_yds_per_att', 'rush_fd', 'penalties', 'penalties_yds', 'pen_fd',
       'score_pct', 'turnover_pct', 'exp_pts_tot', 'ties'],
      dtype='object')
Columns to drop: ['team', 'year']
wins                      int64
losses                    int64
win_loss_perc           float64
points                    int64
points_opp                int64
points_diff               int64
mov                     float64
g                         int64
total_yards               int64
plays_offense             int64
yds_per_play_offense    float64
turnovers                 int64
fumbles_lost              int64
first_down                int64
pass_cmp                  int64
pass_att                  int64
pass_yds                  int64
pass_td                   int64
pass_int                  int64
pass_net_yds_per_att    float64
pass_fd                   int64
rush_att                  int64
rush_yds                  int64
rush_td                   int64
rush_yds_per_att        float64
rush_fd                   int64
penalties                 int64
penalties_yds             int64
pen_fd                    int64
score_pct               float64
turnover_pct            float64
exp_pts_tot             float64
ties                    float64
dtype: object
```




```
                 Feature       Score
12  pass_net_yds_per_att  297.659093
1            total_yards  204.022200
10               pass_td  193.546016
3   yds_per_play_offense  183.885858
6             first_down  177.420983
22          turnover_pct  173.368508
23           exp_pts_tot  164.858126
4              turnovers  156.743036
16               rush_td  151.922786
11              pass_int  136.711731
14              rush_att  102.221582
18               rush_fd   88.983523
9               pass_yds   77.741296
13               pass_fd   65.257350
15              rush_yds   56.432111
5           fumbles_lost   44.410812
2          plays_offense   38.091267
7               pass_cmp   15.759148
21                pen_fd    8.835691
19             penalties    5.767016
17      rush_yds_per_att    4.697230
20         penalties_yds    1.115565
8               pass_att    0.638217
0                      g    0.004808
Linear Regression RMSE: 0.10724015127315736
Linear Regression R^2: 0.7052431508364485
Random Forest RMSE: 0.1208962013586984
Random Forest R^2: 0.6253943010717984
                 Feature  Importance
12  pass_net_yds_per_att    0.334012
14              rush_att    0.108807
22          turnover_pct    0.104472
16               rush_td    0.047844
10               pass_td    0.045416
23           exp_pts_tot    0.029697
18               rush_fd    0.023322
21                pen_fd    0.022422
4              turnovers    0.022152
19             penalties    0.021846
20         penalties_yds    0.021405
15              rush_yds    0.021185
17      rush_yds_per_att    0.020678
6             first_down    0.020395
11              pass_int    0.020163
2          plays_offense    0.019908
1            total_yards    0.019712
7               pass_cmp    0.018327
5           fumbles_lost    0.016483
9               pass_yds    0.014785
13               pass_fd    0.013955
8               pass_att    0.013167
3   yds_per_play_offense    0.012911
0                      g    0.006935
```



In [ ]: