In [1]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
fantasy_data = pd.read_csv('fantasy sports roi calculator data.csv')
```

In [3]:

```
fantasy_data
```

Out[3]:

	Player Name	Matches Played	Avg Fantasy Points	Consistency (%)	Selected %	Credit Cost	Team Points Earned	Prize Pool ($)	Entry Fee ($)	Rank	Winnings ($)
0	Player A	14	25.7	77	54	8.0	1450	2000	20	311	0.0
1	Player B	11	76.8	92	71	9.7	1335	5000	5	424	0.0
2	Player C	18	56.0	67	66	10.0	709	500	10	308	0.0
3	Player D	15	46.2	68	15	8.0	298	500	20	144	0.0
4	Player E	12	25.9	81	37	9.9	352	5000	10	269	0.0
5	Player F	14	37.7	79	37	8.5	1113	500	20	370	0.0
6	Player G	17	38.3	88	53	9.5	1077	500	10	124	0.0
7	Player H	10	62.6	84	93	9.5	537	1000	5	106	0.0
8	Player I	14	58.5	66	39	9.1	1021	500	5	158	0.0
9	Player J	18	70.8	71	71	7.4	1386	2000	10	147	0.0
10	Player K	18	34.5	79	84	10.3	1156	2000	20	145	0.0
11	Player L	15	46.5	45	71	8.3	360	5000	5	120	0.0
12	Player M	12	35.0	79	10	7.7	1000	500	5	319	0.0
13	Player N	11	66.5	81	36	7.2	597	5000	5	275	0.0
14	Player O	15	48.4	91	71	9.4	1015	2000	10	92	100.0
15	Player P	15	36.4	58	86	9.7	1358	500	10	58	25.0
16	Player Q	10	56.2	47	12	7.1	1418	500	20	439	0.0
17	Player R	13	26.7	45	79	9.0	600	5000	20	346	0.0
18	Player S	12	71.3	49	81	7.9	839	2000	20	229	0.0
19	Player T	9	49.7	70	36	9.6	1256	500	10	474	0.0

	Player Name	Matches Played	Avg Fantasy Points	Consistency (%)	Selected %	Credit Cost	Team Points Earned	Prize Pool ($)	Entry Fee ($)	Rank	Winnings ($)
20	Player U	15	46.7	58	18	7.7	887	5000	10	117	0.0
21	Player V	19	76.0	83	71	9.8	659	2000	20	318	0.0
22	Player W	13	65.0	71	46	8.5	1154	5000	20	279	0.0
23	Player X	9	43.0	53	60	10.7	669	1000	10	127	0.0
24	Player Y	19	56.4	59	53	7.6	1245	500	20	393	0.0
25	Player Z	12	53.6	59	33	8.4	949	1000	5	396	0.0

In [4]:

```
x = fantasy_data.iloc[:,1:10].values
y = fantasy_data.iloc[:,10].values
```

In [5]:

```
x
```

```
Out[5]: array([[  14. ,   25.7,   77. ,   54. ,    8. , 1450. , 2000. ,   20. ,
          311. ],
        [  11. ,   76.8,   92. ,   71. ,    9.7, 1335. , 5000. ,    5. ,
          424. ],
        [  18. ,   56. ,   67. ,   66. ,   10. ,  709. ,  500. ,   10. ,
          308. ],
        [  15. ,   46.2,   68. ,   15. ,    8. ,  298. ,  500. ,   20. ,
          144. ],
        [  12. ,   25.9,   81. ,   37. ,    9.9,  352. , 5000. ,   10. ,
          269. ],
        [  14. ,   37.7,   79. ,   37. ,    8.5, 1113. ,  500. ,   20. ,
          370. ],
        [  17. ,   38.3,   88. ,   53. ,    9.5, 1077. ,  500. ,   10. ,
          124. ],
        [  10. ,   62.6,   84. ,   93. ,    9.5,  537. , 1000. ,    5. ,
          106. ],
        [  14. ,   58.5,   66. ,   39. ,    9.1, 1021. ,  500. ,    5. ,
          158. ],
        [  18. ,   70.8,   71. ,   71. ,    7.4, 1386. , 2000. ,   10. ,
          147. ],
        [  18. ,   34.5,   79. ,   84. ,   10.3, 1156. , 2000. ,   20. ,
          145. ],
        [  15. ,   46.5,   45. ,   71. ,    8.3,  360. , 5000. ,    5. ,
          120. ],
        [  12. ,   35. ,   79. ,   10. ,    7.7, 1000. ,  500. ,    5. ,
          319. ],
        [  11. ,   66.5,   81. ,   36. ,    7.2,  597. , 5000. ,    5. ,
          275. ],
        [  15. ,   48.4,   91. ,   71. ,    9.4, 1015. , 2000. ,   10. ,
           92. ],
        [  15. ,   36.4,   58. ,   86. ,    9.7, 1358. ,  500. ,   10. ,
           58. ],
        [  10. ,   56.2,   47. ,   12. ,    7.1, 1418. ,  500. ,   20. ,
          439. ],
        [  13. ,   26.7,   45. ,   79. ,    9. ,  600. , 5000. ,   20. ,
          346. ],
        [  12. ,   71.3,   49. ,   81. ,    7.9,  839. , 2000. ,   20. ,
          229. ],
        [   9. ,   49.7,   70. ,   36. ,    9.6, 1256. ,  500. ,   10. ,
          474. ],
        [  15. ,   46.7,   58. ,   18. ,    7.7,  887. , 5000. ,   10. ,
          117. ],
        [  19. ,   76. ,   83. ,   71. ,    9.8,  659. , 2000. ,   20. ,
          318. ],
        [  13. ,   65. ,   71. ,   46. ,    8.5, 1154. , 5000. ,   20. ,
          279. ],
        [   9. ,   43. ,   53. ,   60. ,   10.7,  669. , 1000. ,   10. ,
          127. ],
        [  19. ,   56.4,   59. ,   53. ,    7.6, 1245. ,  500. ,   20. ,
          393. ],
        [  12. ,   53.6,   59. ,   33. ,    8.4,  949. , 1000. ,    5. ,
          396. ]])
```

```
In [6]: y
```

```
Out[6]: array([  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0., 100.,  25.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.])
```

In [7]:

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=
```

In [8]:

```
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

Out[8]:

```
▾        RandomForestRegressor
RandomForestRegressor(random_state=42)
```

In [9]:

```
y_pred = model.predict(X_test)
```

In [10]:

```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared Score: {r2:.2f}")
```

```
Mean Squared Error: 8.01
R-squared Score: 0.00
```

In [12]:

```
features = ['Matches Played', 'Avg Fantasy Points', 'Consistency (%)',
            'Selected %', 'Credit Cost', 'Team Points Earned', 'Prize Pool ($)', 'Entr
```

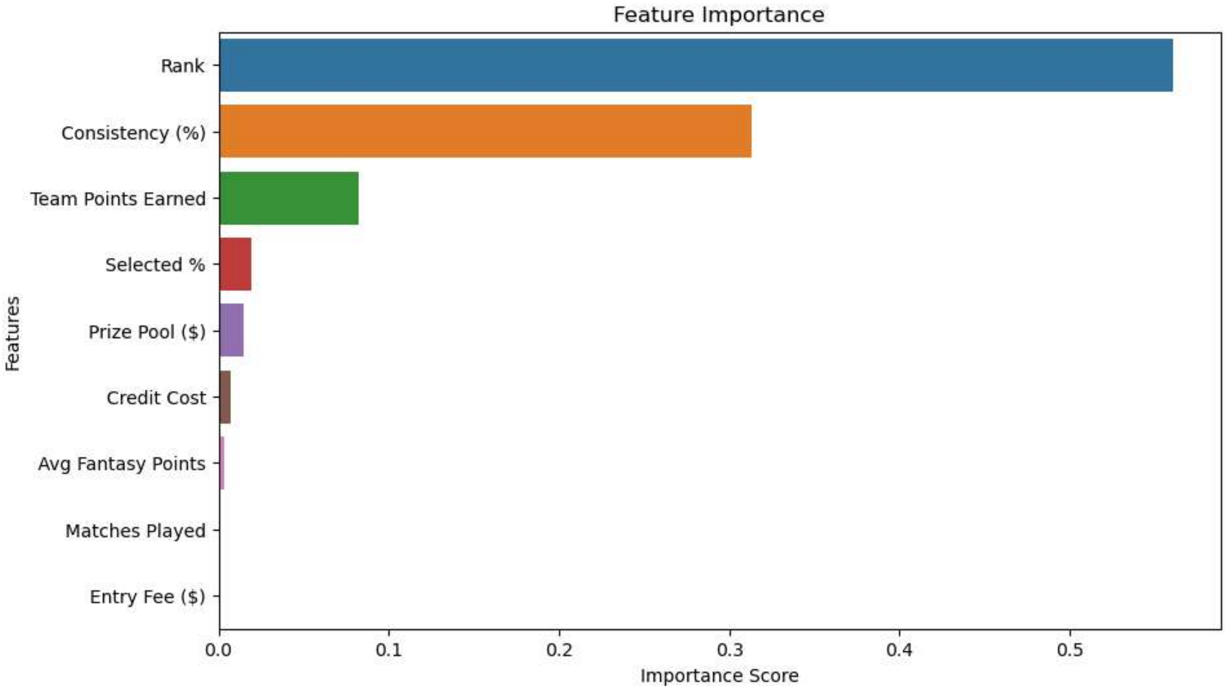
In [13]:

```
importances = model.feature_importances_
feature_importance = pd.Series(importances, index=features).sort_values(ascending=Fals
```

In [14]:

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10,6))
sns.barplot(x=feature_importance, y=feature_importance.index)
plt.title('Feature Importance')
plt.xlabel('Importance Score')
plt.ylabel('Features')
plt.show()
```


Feature Importance
Rank
Consistency (%)
Team Points Earned
Selected %
Prize Pool ($)
Credit Cost
Avg Fantasy Points
Matches Played
Entry Fee ($)
Features
0.0
0.1
0.2
0.3
0.4
0.5
Importance Score

```
In [ ]:
```