In [1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from tabulate import tabulate
import warnings
```

```
C:\Users\Prave\anaconda3\lib\site-packages\pandas\core\arrays\masked.py:60: UserWarning: Pandas requires version
'1.3.6' or newer of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
```

In [2]:

```python
df_opponent = pd.read_excel("C://Users//Prave//Downloads//basketball_LTuw.xlsx",sheet_name="GamebyGame_Oppents")
```

In [3]:

```python
# Split FGM/A into FGM and FGA
df_opponent[['FGM', 'FGA']] = df_opponent['FGM/A'].str.split('-', expand=True).astype(int)

# Split 3FG/A into 3FGM and 3FGA
df_opponent[['3FGM', '3FGA']] = df_opponent['3FG/A'].str.split('-', expand=True).astype(int)

# Split FTM/A into FTM and FTA
df_opponent[['FTM', 'FTA']] = df_opponent['FTM/A'].str.split('-', expand=True).astype(int)

# Drop the original combined columns if needed
df_opponent.drop(columns=['FGM/A', '3FG/A', 'FTM/A'], inplace=True)
```

In [4]:

```python
df_ltu = pd.read_excel("C://Users//Prave//Downloads//basketball_LTuw.xlsx",sheet_name="Overall_GameByGameTeamstats"
```

In [5]:

```python
# Strip leading/trailing whitespaces in 'Opponent' columns
df_ltu['Opponent'] = df_ltu['Opponent'].str.strip()
df_opponent['Opponent'] = df_opponent['Opponent'].str.strip()
```

In [6]:

```python
# Convert 'Date' columns to datetime format
df_ltu['Date'] = pd.to_datetime(df_ltu['Date'])
df_opponent['Date'] = pd.to_datetime(df_opponent['Date'])
```

In [7]:
```python
# Now attempt to merge again
df_merged = pd.merge(df_ltu, df_opponent, on=['Opponent', 'Date'], suffixes=('_ltu', '_opponent'))
```

In [8]:
```python
# Find rows in df_ltu that have no match in df_opponent
unmatched_ltu = df_ltu[~df_ltu[['Opponent', 'Date']].apply(tuple, 1).isin(df_opponent[['Opponent', 'Date']].apply(t
print("Unmatched rows in df_ltu:")
#print(unmatched_ltu)
```

```
Unmatched rows in df_ltu:
```

In [9]:
```python
# Remove 'vs ' from the 'Opponent' column in df_ltu
df_ltu['Opponent'] = df_ltu['Opponent'].str.replace(r'^vs\s+', '', regex=True)

# Check the updated dataframe to ensure the 'vs' prefix is removed
print(df_ltu['Opponent'].head())
```

```
0    Mount Vernon Nazarene
1             Indiana East
2            at Rio Grande
3                Ave Maria
4                    Grace
Name: Opponent, dtype: object
```

In [10]:
```python
# Now attempt to merge again
df_merged = pd.merge(df_ltu, df_opponent, on=['Opponent', 'Date'], suffixes=('_ltu', '_opponent'))
```

In [11]:
```python
# Convert 'W/L' to binary format: Win = 1, Loss = 0
df_merged['W/L_ltu'] = df_merged['W/L_ltu'].apply(lambda x: 1 if x == 'W' else 0)
df_merged['W/L_opponent'] = df_merged['W/L_opponent'].apply(lambda x: 1 if x == 'W' else 0)
```

In [12]:
```
df_merged.columns
```

Out[12]:
```
Index(['Opponent', 'Date', 'Score_ltu', 'W/L_ltu', 'FGM/A', 'PCT_ltu', '3FG/A',
       'PCT_1_ltu', 'FTM/A', 'PCT_2_ltu', 'OFF_ltu', 'DEF_ltu', 'TOT_ltu',
       'AVG_ltu', 'PF_ltu', 'AST_ltu', 'TO_ltu', 'BLK_ltu', 'STL_ltu',
       'PTS_ltu', 'AVG_3_ltu', 'Year_ltu', 'Score_opponent', 'W/L_opponent',
       'PCT_opponent', 'PCT_1_opponent', 'PCT_2_opponent', 'OFF_opponent',
       'DEF_opponent', 'TOT_opponent', 'AVG_opponent', 'PF_opponent',
       'AST_opponent', 'TO_opponent', 'BLK_opponent', 'STL_opponent',
       'PTS_opponent', 'AVG_3_opponent', 'Year_opponent', 'FGM', 'FGA', '3FGM',
       '3FGA', 'FTM', 'FTA'],
      dtype='object')
```

In [13]:
```
df_merged.head()
```

Out[13]:

	Opponent	Date	Score_ltu	W/L_ltu	FGM/A	PCT_ltu	3FG/A	PCT_1_ltu	FTM/A	PCT_2_ltu	...	STL_opponent	PTS_opponent	AVG_3_opp
0	Mount Vernon Nazarene	2021-10-22	71-73	0	27-71	0.380	8-23	0.348	9-12	0.750	...	6	73	
1	Indiana East	2021-10-29	59-75	0	23-76	0.303	1-15	0.067	12-14	0.857	...	13	75	
2	Ave Maria	2021-11-06	73-56	1	19-54	0.352	7-25	0.280	28-35	0.800	...	9	56	
3	Grace	2021-11-07	60-75	0	21-51	0.412	7-18	0.389	11-13	0.846	...	16	75	
4	Concordia (MI)	2021-11-10	67-53	1	21-56	0.375	5-13	0.385	20-24	0.833	...	6	53	

5 rows × 45 columns

In [37]:

```python
df = pd.DataFrame(df_merged)

# Calculate the differences
df['PCT_Diff'] = df['PCT_ltu'] - df['PCT_opponent']
df['PCT_1_Diff'] = df['PCT_1_ltu'] - df['PCT_1_opponent']
df['PCT_2_Diff'] = df['PCT_2_ltu'] - df['PCT_2_opponent']
df['OFF_Diff'] = df['OFF_ltu'] - df['OFF_opponent']
df['DEF_Diff'] = df['DEF_ltu'] - df['DEF_opponent']
df['AVG_Diff'] = df['AVG_ltu'] - df['AVG_opponent']
df['PF_Diff'] = df['PF_ltu'] - df['PF_opponent']
df['AST_Diff'] = df['AST_ltu'] - df['AST_opponent']
df['TO_Diff'] = df['TO_ltu'] - df['TO_opponent']
df['BLK_Diff'] = df['BLK_ltu'] - df['BLK_opponent']
df['STL_Diff'] = df['STL_ltu'] - df['STL_opponent']
```

In [92]:

```python
df.head()
```

Out[92]:

	Opponent	Date	Score_ltu	W/L_ltu	FGM/A	PCT_ltu	3FG/A	PCT_1_ltu	FTM/A	PCT_2_ltu	...	PCT_1_Diff	PCT_2_Diff	OFF_Diff	DEF_Dif
0	Mount Vernon Nazarene	2021-10-22	71-73	0	27-71	0.380	8-23	0.348	9-12	0.750	...	-0.231	-0.139	4	-1
1	Indiana East	2021-10-29	59-75	0	23-76	0.303	1-15	0.067	12-14	0.857	...	-0.150	0.232	11	-
2	Ave Maria	2021-11-06	73-56	1	19-54	0.352	7-25	0.280	28-35	0.800	...	0.224	0.271	-5	1
3	Grace	2021-11-07	60-75	0	21-51	0.412	7-18	0.389	11-13	0.846	...	0.103	0.260	-14	-
4	Concordia (MI)	2021-11-10	67-53	1	21-56	0.375	5-13	0.385	20-24	0.833	...	0.145	0.404	-7	1

5 rows × 56 columns

In [38]:
```python
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
```

In [39]:
```python
pca_features = ['PCT_Diff', 'PCT_1_Diff', 'PCT_2_Diff',
                'OFF_Diff', 'DEF_Diff', 'AVG_Diff',
                'PF_Diff', 'AST_Diff', 'TO_Diff', 'BLK_Diff', 'STL_Diff']
```

In [40]:
```python
df_pca = df[pca_features]
```

In [41]:
```python
# Standardize the features
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df_pca)
```

In [53]:
```python
# Create a PCA model
pca = PCA(n_components=len(pca_features))  # you can also set to a smaller number
pca.fit(df_scaled)
```

Out[53]:
```
PCA
PCA(n_components=11)
```

In [54]:
```python
# Explained variance ratio
explained_variance = pca.explained_variance_ratio_
# Cumulative explained variance
cumulative_variance = explained_variance.cumsum()
```

In [55]:
```python
# Create a DataFrame for PCA components
pca_components = pd.DataFrame(pca.components_, columns=pca_features)
```

In [56]:

```
# Display explained variance
print("Explained Variance Ratios:", explained_variance)
print("Cumulative Explained Variance:", cumulative_variance)
```

```
Explained Variance Ratios: [0.28013896 0.21440715 0.12389004 0.10870809 0.07462294 0.06285699
 0.04971499 0.03988684 0.02795411 0.01399176 0.00382813]
Cumulative Explained Variance: [0.28013896 0.49454611 0.61843616 0.72714425 0.80176718 0.86462418
 0.91433916 0.954226   0.98218011 0.99617187 1.        ]
```

In [59]:

```
# Step 4: Determine the number of components to reach 90% explained variance
n_components = np.argmax(cumulative_variance >= 0.90) + 1
```

In [60]:

```
print(f'Number of components to explain 90% variance: {n_components}')
```

```
Number of components to explain 90% variance: 7
```

In [82]:

```
# Step 6: Print the PCA components that explain the variance
components_df = pd.DataFrame(pca.components_[:n_components],
                              columns=pca_features)
```

In [86]:
```python
# Display the PCA components
print('PCA Components explaining 90% variance:')
print(tabulate(components_df.T,headers='keys', tablefmt='grid'))
```

PCA Components explaining 90% variance:

	0	1	2	3	4	5	6
PCT_Diff	0.47203	0.138543	-0.0839805	0.152931	-0.257331	0.0374307	0.421478
PCT_1_Diff	0.423914	0.131817	0.13383	0.328205	-0.107671	0.143171	-0.182205
PCT_2_Diff	0.106717	-0.143328	-0.585796	0.0678183	0.66552	0.301019	-0.143294
OFF_Diff	-0.079076	0.40907	-0.0470156	-0.561633	0.102437	-0.207192	0.231727
DEF_Diff	0.492253	0.0270308	-0.198245	-0.226844	0.0485648	0.121153	0.43418
AVG_Diff	0.144962	0.406818	0.18732	-0.428093	0.199199	0.071547	-0.342823
PF_Diff	-0.0791889	0.0416267	0.656045	0.219044	0.509956	0.233699	0.354634
AST_Diff	0.330826	0.38298	0.112018	0.151808	0.0180457	0.055092	-0.461044
TO_Diff	0.261465	-0.430264	0.292293	-0.349506	0.112992	0.134365	-0.0042997
BLK_Diff	0.306478	-0.201046	0.0702149	0.110787	0.324409	-0.852333	-0.0647565
STL_Diff	-0.19623	0.485144	-0.148227	0.329676	0.219885	-0.15029	0.257583

Feature Descriptions, Their Impact on Winning and Losing, and LTU's Weaknesses

PCT_Diff (Percentage Difference):

Description: Indicates the shooting percentage difference between LTU and their opponents. Impact: A positive value (e.g., 0.47203) suggests LTU shot better, increasing their chances of winning. However, a negative value (-0.257331) indicates games where LTU performed poorly in shooting, which can lead to losses.

PCT_1_Diff (First Percentage Difference):

Description: Measures a specific shooting stats difference. Impact: A strong positive value (e.g., 0.328205) correlates with winning, while a negative value (-0.182205) reflects missed scoring opportunities, showing LTU's weakness in certain shooting situations.

PCT_2_Diff (Second Percentage Difference):

Description: Compares another set of shooting stats. Impact: Values like 0.66552 indicate effective shooting; however, a significant negative value (-0.585796) highlights LTU's struggles, showing they failed to capitalize on scoring chances in those games.

OFF_Diff (Offensive Difference):

Description: Represents the difference in points scored per possession. Impact: A negative value (-0.561633) suggests LTU struggled offensively in some games, which often leads to losses due to insufficient scoring.

DEF_Diff (Defensive Difference):

Description: Measures points allowed per possession. Impact: A positive value (0.492253) indicates strong defense, but a negative value (-0.198245) shows instances where LTU's defense faltered, allowing opponents to score easily and leading to potential losses.

AVG_Diff (Average Difference):

Description: Captures overall performance across metrics. Impact: A positive value (0.406818) is favorable, but a negative value (-0.428093) indicates inconsistency, which could lead to losses due to poor overall performance.

PF_Diff (Personal Fouls Difference):

Description: Shows the difference in fouls committed. Impact: A significant positive value (0.656045) indicates LTU committed many fouls, leading to more free throw opportunities for opponents, which is a critical weakness.

AST_Diff (Assists Difference):

Description: Measures the difference in assists. Impact: A positive value (0.38298) indicates good teamwork, but LTU has instances with negative values (e.g., -0.461044), reflecting poor ball movement and fewer scoring opportunities, contributing to losses.

TO_Diff (Turnovers Difference):

Description: Indicates the difference in turnovers. Impact: A negative value (-0.430264) signifies that LTU committed more turnovers than opponents in some games, which can lead to missed scoring chances and losses

BLK_Diff (Blocks Difference):

Description: Captures the difference in blocked shots. Impact: A positive value (0.324409) indicates effective defense. However, LTU's negative value (-0.852333) in some games highlights severe defensive lapses, allowing opponents to score easily

STL_Diff (Steals Difference):

Description: Measures the difference in steals. Impact: A positive value (0.485144) suggests effective defensive plays. Yet, negative values (e.g., -0.19623) reflect a lack of defensive intensity, which can lead to losses.

Summary These metrics reveal LTU's performance trends over the past five years. Positive differences indicate successful areas contributing to wins, while negative differences pinpoint weaknesses that can lead to losses. For instance, LTU's high PF_Diff (0.656045) shows they often commit many fouls, giving opponents more free throw opportunities. Similarly, a significant TO_Diff (-0.430264) reveals turnover issues, affecting scoring chances. Addressing these weaknesses—such as improving shooting efficiency, reducing turnovers, and minimizing fouls—can help enhance LTU's chances of winning in future games.

In [104]:
```
components_df
```

Out[104]:

	PCT_Diff	PCT_1_Diff	PCT_2_Diff	OFF_Diff	DEF_Diff	AVG_Diff	PF_Diff	AST_Diff	TO_Diff	BLK_Diff	STL_Diff
0	0.472030	0.423914	0.106717	-0.079076	0.492253	0.144962	-0.079189	0.330826	0.261465	0.306478	-0.196230
1	0.138543	0.131817	-0.143328	0.409070	0.027031	0.406818	0.041627	0.382980	-0.430264	-0.201046	0.485144
2	-0.083980	0.133830	-0.585796	-0.047016	-0.198245	0.187320	0.656045	0.112018	0.292293	0.070215	-0.148227
3	0.152931	0.328205	0.067818	-0.561633	-0.226844	-0.428093	0.219044	0.151808	-0.349506	0.110787	0.329676
4	-0.257331	-0.107671	0.665520	0.102437	0.048565	0.199199	0.509956	0.018046	0.112992	0.324409	0.219885
5	0.037431	0.143171	0.301019	-0.207192	0.121153	0.071547	0.233699	0.055092	0.134365	-0.852333	-0.150290
6	0.421478	-0.182205	-0.143294	0.231727	0.434180	-0.342823	0.354634	-0.461044	-0.004300	-0.064757	0.257583

In [105]:

```python
# Define a function to color the background based on the value
def color_heatmap(val):
    color = 'background-color: '
    if val < 0:
        color += 'lightcoral;'  # Red for negative values
    elif val < 0.5:
        color += 'lightyellow;'  # Yellow for lower values
    else:
        color += 'lightgreen;'  # Green for higher values
    return color

# Apply the style to the DataFrame
styled_df = components_df.style.applymap(color_heatmap)

# Display the styled DataFrame
styled_df
```

```
C:\Users\Prave\AppData\Local\Temp\ipykernel_17912\2392187783.py:13: FutureWarning: Styler.applymap has been deprec
ated. Use Styler.map instead.
  styled_df = components_df.style.applymap(color_heatmap)
```

Out[105]:

	PCT_Diff	PCT_1_Diff	PCT_2_Diff	OFF_Diff	DEF_Diff	AVG_Diff	PF_Diff	AST_Diff	TO_Diff	BLK_Diff	STL_Diff
0	0.472030	0.423914	0.106717	-0.079076	0.492253	0.144962	-0.079189	0.330826	0.261465	0.306478	-0.196230
1	0.138543	0.131817	-0.143328	0.409070	0.027031	0.406818	0.041627	0.382980	-0.430264	-0.201046	0.485144
2	-0.083980	0.133830	-0.585796	-0.047016	-0.198245	0.187320	0.656045	0.112018	0.292293	0.070215	-0.148227
3	0.152931	0.328205	0.067818	-0.561633	-0.226844	-0.428093	0.219044	0.151808	-0.349506	0.110787	0.329676
4	-0.257331	-0.107671	0.665520	0.102437	0.048565	0.199199	0.509956	0.018046	0.112992	0.324409	0.219885
5	0.037431	0.143171	0.301019	-0.207192	0.121153	0.071547	0.233699	0.055092	0.134365	-0.852333	-0.150290
6	0.421478	-0.182205	-0.143294	0.231727	0.434180	-0.342823	0.354634	-0.461044	-0.004300	-0.064757	0.257583

In [ ]: