In [1]:
```python
import pandas as pd
import warnings
import numpy as np
import os
from matplotlib import pyplot
import seaborn as sns
import matplotlib.pyplot as plt
warnings.filterwarnings('ignore')
os.chdir("E:\Ginu_StudyMaterials\Sem2\MachineLearning")
```

In [2]:
```python
# reading and printing dataset which is saved after doing feature selection
data = pd.read_csv("data.csv", encoding = 'unicode_escape')
data
```

Out[2]:

	season	playoff	score1	score2	elo1_pre	elo2_pre	elo1_post	elo2_post	prob1	is_home1	...	name2_New York Liberty	name2_Orlando Miracle	name2_Phoenix Mercury	name2_Portland Fire	name2_Sacrament Monarch
0	2019	1	89	78	1684	1634	1692	1627	0.718	1	...	0	0	0	0	
1	2019	1	78	89	1634	1684	1627	1692	0.282	0	...	0	0	0	0	
2	2019	1	86	90	1693	1626	1684	1634	0.476	0	...	0	0	0	0	
3	2019	1	90	86	1626	1693	1634	1684	0.524	1	...	0	0	0	0	
4	2019	1	94	81	1671	1648	1693	1626	0.399	0	...	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
10483	1997	0	73	61	1500	1500	1521	1479	0.387	0	...	0	0	0	0	
10484	1997	0	67	57	1500	1500	1519	1481	0.387	0	...	0	0	0	0	
10485	1997	0	57	67	1500	1500	1481	1519	0.613	1	...	1	0	0	0	
10486	1997	0	61	73	1500	1500	1479	1521	0.613	1	...	0	0	0	0	
10487	1997	0	56	76	1500	1500	1470	1530	0.613	1	...	0	0	0	0	

10488 rows × 92 columns

In [3]:
```python
# transforming data using standard scaler
# here is_home1 is the response variable I selected.

from sklearn.preprocessing import StandardScaler
from numpy import set_printoptions

X= data.copy()
X.drop(columns=['is_home1'], inplace=True)
X = X.values
Y= data['is_home1']
Y=Y.values

scaler=StandardScaler()
knn_data = scaler.fit_transform(X)

from sklearn.model_selection import train_test_split
Ftrain,Ftest,Ltrain,Ltest=train_test_split(X,Y, test_size=0.20, random_state=22)

# 103 is sq.root of n i.e.. sq.root of 10488. So applying KNN with k=103.

from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=103)
model.fit(Ftrain, Ltrain)
model.score(Ftest,Ltest)
```

Out[3]: 0.5376549094375596

Accuracy is 50% with 103 neighbors.

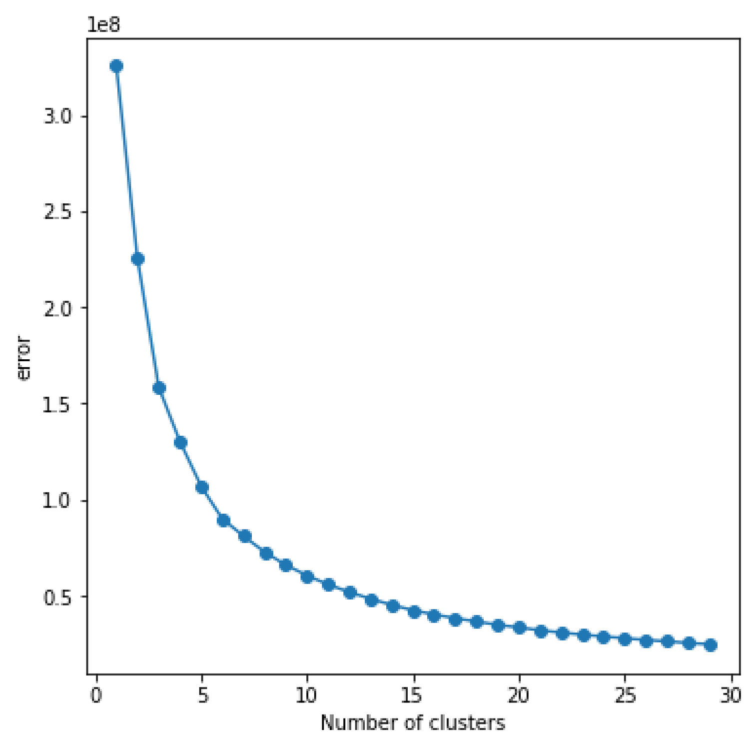
103 is sq.root of n i.e.. sq.root of 10488.

In [4]:
```python
## kmeans clustering with 2 clusters which is based on the 2 components from PCA
from sklearn.cluster import KMeans
Kmean = KMeans(n_clusters=2)
Kmean.fit(X)
Kmeans = Kmean.predict(X)
```

In [5]:
```python
# plotting elbow graph
vals=[]
list_k= list(range(1,30))

for k in list_k:
    km = KMeans(n_clusters=k)
    km.fit(X)
    vals.append(km.inertia_)

# Plot sse against k
plt.figure(figsize=(6, 6))
plt.plot(list_k, vals, '-o')
plt.xlabel('Number of clusters')
plt.ylabel('error')
plt.show()
```



In [6]:
```python
# MODELLING WITH 6 SINCE CURVE STARTS AT 6 (refer Kmean file for elbow graph)
model = KNeighborsClassifier(n_neighbors=6)
model.fit(Ftrain, Ltrain)
model.score(Ftest,Ltest)
```

Out[6]: 0.6072449952335558

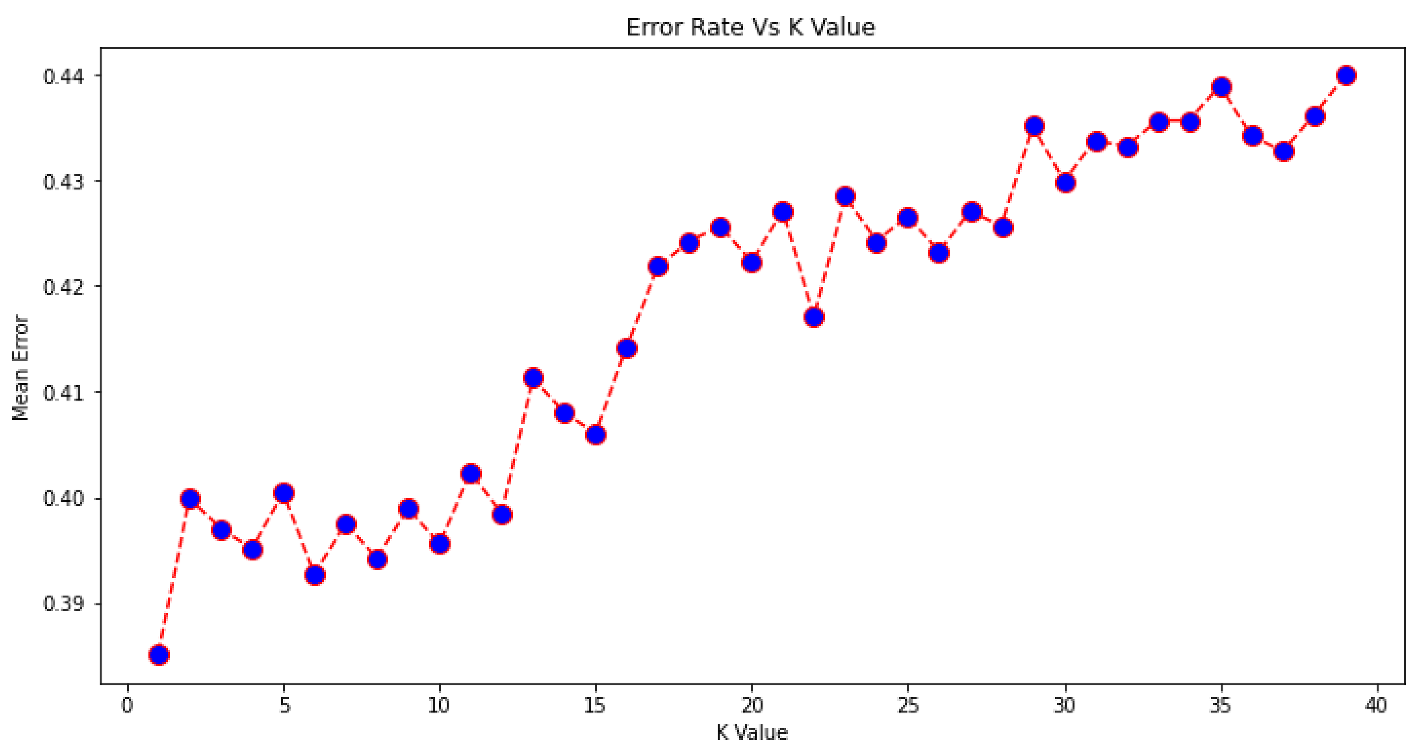
Accuracy increased to 60 % with 6 neighbors.

6 is the number from the Elbow curve where the curve starts.

In [7]:
```python
error = []

# Calculating error for K values between 1 and 40
for i in range(1, 40):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(Ftrain, Ltrain)
    pred_i = knn.predict(Ftest)
    error.append(np.mean(pred_i != Ltest))
```

In [8]:
```python
# plotting error rate vs K value
plt.figure(figsize=(12, 6))
plt.plot(range(1, 40), error, color='red', linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
plt.title('Error Rate Vs K Value')
plt.xlabel('K Value')
plt.ylabel('Mean Error')
plt.show()
```



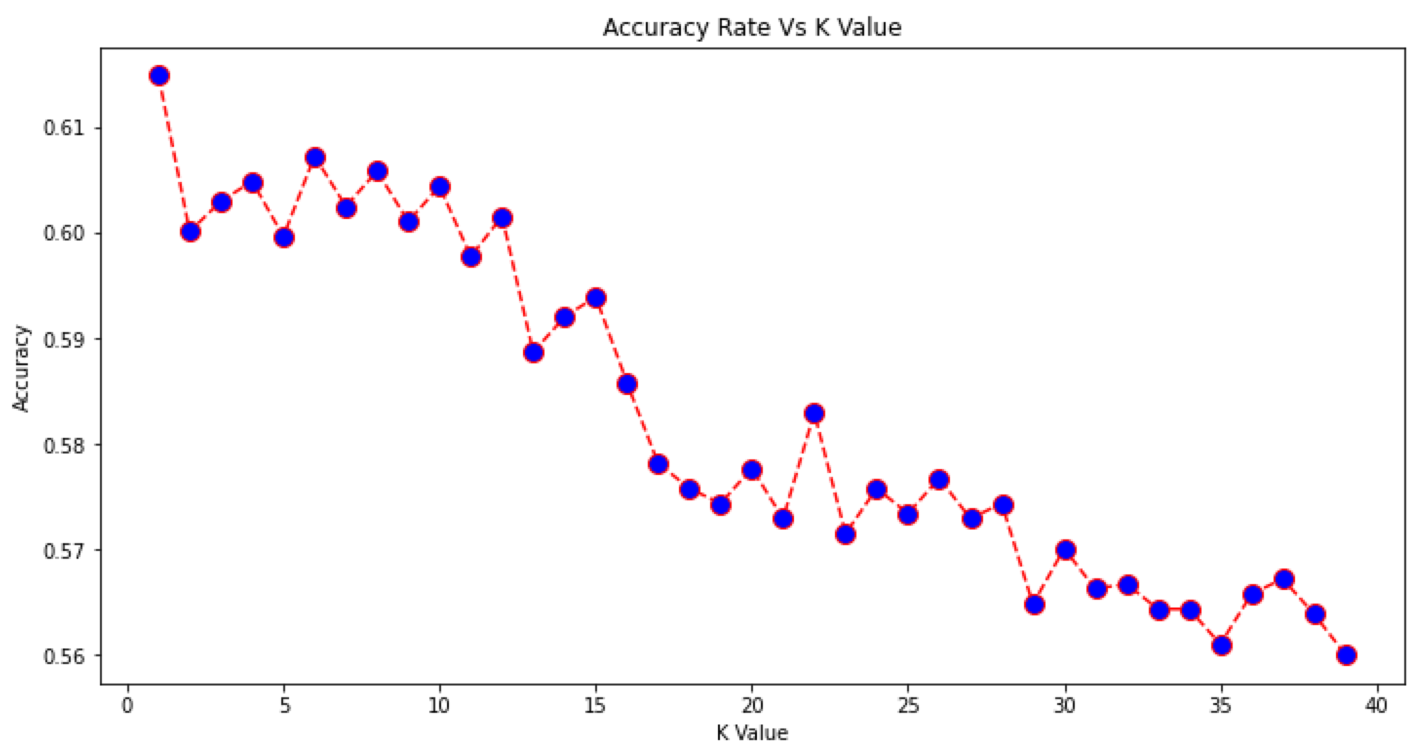
As the k value incraeses the mean error seems to be increasing.

In [9]:
```python
acc = []

# Calculating error for K values between 1 and 40
for i in range(1, 40):
    knn_model = KNeighborsClassifier(n_neighbors=i)
    knn_model.fit(Ftrain, Ltrain)
    pred_i = knn_model.predict(Ftest)
    acc.append(knn_model.score(Ftest,Ltest))
```

In [10]:
```python
# plotting accuracy vs k value

plt.figure(figsize=(12, 6))
plt.plot(range(1, 40), acc, color='red', linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
plt.title('Accuracy Rate Vs K Value')
plt.xlabel('K Value')
plt.ylabel('Accuracy')
plt.show()
```



From the plot its showing that as the K value increses, the accuracy is decresing.

In [ ]: