In [ ]:

```
#*****************************************************************************
#
#       Problem : To Predict How many points NBA players scores in 2013-2014 session
#                 using Knn Algoritm
#
#*****************************************************************************
# Step0: To import packages and load Data
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.neighbors import KNeighborsRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score,mean_squared_error
from sklearn.model_selection import cross_val_score

import seaborn as sns

# Loading the Data by using pandas

player_Data = pd.read_csv("nba_2013.csv")

print(" The NBA Player Data is  \n",'-'*80)
print(player_Data.head())


```

In [ ]:
```
# Step1: To Analyse the Data

# To know the Shape of the Data , number of rows and columns
print(player_Data.shape)
print('-'*80)

# to know the information of each feature and data types
print(player_Data.describe())
print('-'*80)

print(player_Data.info())
print('-'*80)

# to check if any values missing or nulls in the Dataset

print(player_Data.isnull().sum())
print('-'*80)
```

In [ ]:
```
# Setp2: preprocessing the Data
# We observe from step2 , that , there are nulls in few records so we will impute and replace by mean


player_Data = player_Data.fillna(player_Data.mean())
# to check if all thr nulls are replaced after imputing

# converting Categorical fields to numerical by using LabelEncoder
le = LabelEncoder()

player_Data['pos'] = le.fit_transform(player_Data['pos'].astype(str))
player_Data['bref_team_id'] = le.fit_transform(player_Data['bref_team_id'].astype(str))

print(player_Data.isnull().sum())
print(player_Data.columns)
```

In [ ]:
```
# Step3: Visualisation of Data

features= ['pos', 'age', 'bref_team_id','g', 'gs', 'mp', 'fg', 'fga', 'fg.',
       'x3p', 'x3pa', 'x3p.', 'x2p', 'x2pa', 'x2p.', 'efg.', 'ft', 'fta',
       'ft.', 'orb', 'drb', 'trb', 'ast', 'stl', 'blk', 'tov', 'pf']
print(features)
print('-'*80)
print("Visualising the Atrributes of the DataSet with respect to the Point scored by Player")
y = 'pts'
X = features
i = 1

columncnt = len(features)
while i < (columncnt-2):
    sns.pairplot(player_Data,x_vars=X[i:i+5],y_vars=y,kind="reg")
    i = i+5

```

In [ ]:
```
# Step4: Train test split of Data
features = ['g', 'gs', 'mp', 'fg', 'fga', 'x3p', 'x3pa','x2p', 'x2pa', 'ft', 'fta', 'orb', 'drb', 'trb', 'ast',
            'stl', 'blk', 'tov', 'pf']

X = player_Data[features]
y = player_Data[y]

y = y.values.reshape(-1,1)

# splitting the data into training and test Data set for 30 percent testing Data
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.30,random_state=2)
print(X_train.shape)
print(X_test.shape)

print(y_train.shape)
print(y_test.shape)


```

In [ ]:

```
# Step5: Model Building - Using Knn algorithm , finding the least K value

myList = list(range(1,25))

 # subsetting just the odd ones
#neighbors = filter(lambda x: x % 2 != 0, myList)

print(" Finding the Accuracy and error for k values\n",'-'*80)


errlist = []
for k in range(1,20):
    knn = KNeighborsRegressor(n_neighbors=k)
    knn.fit(X_train,y_train)
    y_pred = knn.predict(X_test)
    acc = r2_score(y_test,y_pred)
    err = np.sqrt(mean_squared_error(y_test,y_pred))
    errlist.append((k,err,acc))
    print(" k = {} : The Accuracy score = {} and error is = {} ".format(k,acc,err))


```

In [ ]:

```
# Step6: Model Evaluation
# visualising the error with respect to k value to identify the least k value to get best accuracy

# unzipping the K and error lists to separate lists
k, err,acc = zip(*errlist)

minerror = min(err)
for erritm in errlist:
    if (erritm[1] == minerror):
        kmin = round(erritm[0],3)
        accmax = round(erritm[2],3)
        break

plt.plot(list(k),list(err))
plt.ylabel('Error')
plt.xlabel('K')
print(" From the Graph of Error for various values of K,\n  we observe  error is least = {} at k = {} ".format(round(
print('-'*80)
print(" The scores of NBA players can be predicted by Knn at K=4 at {} % ".format((accmax*100)))
```