```
# Data Preprocessing


# In the data, the teams seeds have W = East, X = Midwest, Y = South, Z = West


from __future__ import division
import pandas as pd
import numpy as np
import os.path
import math
import collections

reg_season_compact_pd = pd.read_csv('Data/KaggleData/RegularSeasonCompactResults.csv')
teams_pd = pd.read_csv('Data/KaggleData/Teams.csv')
tourney_compact_pd = pd.read_csv('Data/KaggleData/NCAATourneyCompactResults.csv')
conference_pd = pd.read_csv('Data/KaggleData/Conference.csv')
tourney_results_pd = pd.read_csv('Data/KaggleData/TourneyResults.csv')
tourney_seeds_pd = pd.read_csv('Data/KaggleData/NCAATourneySeeds.csv')
team_conferences_pd = pd.read_csv('Data/KaggleData/TeamConferences.csv')

teamList = teams_pd['TeamName'].tolist()
NCAAChampionsList = tourney_results_pd['NCAA Champion'].tolist()


allStats = [[[]for _ in range(68)]for _ in range(26)]

def normalizeInput(arr):
    for i in range(arr.shape[1]):
        minVal = min(arr[:,i])
        maxVal = max(arr[:,i])
        arr[:,i] =  (arr[:,i] - minVal) / (maxVal - minVal)
    return arr

def populateTeams():
    j = 0
    temp = 1993
    for i in range(513, 2218):
        currentYear = tourney_seeds_pd.loc[i].at['Season']
        if(temp != currentYear):
            j = 0
        allStats[currentYear-1993][j].append(tourney_seeds_pd.loc[i].at['TeamID'])
        temp = currentYear
        j += 1

def populateSeeds():
    j = 0
    temp = 1993
    for i in range(513, 2218):
        currentYear = tourney_seeds_pd.loc[i].at['Season']
        if(temp != currentYear):
            j = 0
        allStats[currentYear-1993][j].append(1-(int(tourney_seeds_pd.loc[i].at['Seed'][1:3])-1)/15)
        temp = currentYear
        j += 1
```

```
populateTeams()
populateSeeds()
print(allStats)
```