

Desarrollo de un modelo para la predicción de los Juegos Panamericanos de Lima 2019 utilizando Machine Learning

Javier Alejandro Oramas López C-121
Ana Paula Argüelles Terrón C-111
Mauro J. Bolado Vizoso C-111
Daniel Alejandro Cárdenas Cabrera C-113

16 de Julio de 2019

1. Introducción

El presente artículo aborda el tema de realizar una predicción a partir de un método de Machine Learning para obtener un pronóstico aproximado con vista a los Juegos Panamericanos que tendrán lugar en Lima este 2019.

Para ello se presentará a continuación conceptos introductorios al tema sin los cuales fuese imposible el total entendimiento de lo desarrollado.

Inteligencia:Facultad intelectual con que se captan y forman ideas y se crean relaciones.

Aprendizaje: Aprendizaje es un cambio en la disposición o capacidad humana, que persiste durante un tiempo y no pueden atribuirse simplemente a los procesos de crecimiento biológico.(Gagné, 1987).

Inteligencia Artificial: La Inteligencia Artificial es la ciencia cibernética que se encarga de la elaboración de sistemas que "le den una solución inteligente a los problemas", dicho de otra manera los sistemas inteligentes solucionan problemas que de ser resueltos por un humano requirirían un comportamiento inteligente. La Inteligencia Artificial, con todas sus tendencias, es una de las disciplinas más prometedoras de las ciencias de la computación y a la que varios científicos están dedicando ingentes esfuerzos en todo el mundo.

Los científicos que han abordado y abordan este campo se encuentran con los problemas que ello conlleva:

1. - Los computadores no pueden manejar (no contienen) verdaderos significados.
2. - Los computadores no tienen autoconciencia (emociones, sociabilidad, etc.).
3. - Un computador solo puede hacer aquello para lo que está programado.
4. - Las máquinas no pueden pensar realmente.

Problemas a los cuales también se enfrenta este trabajo.

2. Desarrollo

Ante todo se procedió a la recopilación de los datos necesarios para el aprendizaje del programa, de acuerdo a los objetivos requeridos. Estos son esenciales para cualquier trabajo con Machine Learning o Artificial Intelligence (A.I.) por sus siglas en inglés, ya que sin una buena selección y manejo de los datos no se puede llegar a resultados fiables.

Una vez obtenidos los datos a procesar se impuso la necesidad de establecer el algoritmo a utilizar para su procesamiento.

Cuando se pretende predecir un resultado, las personas con conocimientos del tema, sin dudas, en primera instancia piensan en utilizar un algoritmo básico, "Lineal Regression", que aproxima el resultado trazando la recta que mejor describa el comportamiento de los mismos, aportando una solución al problema. No obstante, el resultado obtenido por este método, en algunos casos, como en el nuestro, es bastante impreciso dado que la gráfica producida por los datos no describe una función lineal, en discordancia con el algoritmo antes mencionado. Por ello en nuestro caso optamos por implementar un algoritmo utilizando "Polynomial Regression".

Para el desarrollo de dichos algoritmos se utilizaron las siguientes librerías de Python:

1. - pandas
2. - sklearn
3. - matplotlib
4. - numpy

Pandas es la librería que permite importar los datos almacenados en un archivo cuya extensión sea .csv para su manejo desde el programa.

Sklearn contiene ya implementados los algoritmos de Machine Learning previamente mencionados y proporciona muchas facilidades con el manejo de datos para su procesamiento.

Matplotlib permite graficar los datos para una mejor visualización de estos.

Numpy permite convertir las listas de datos en array para un mejor manejo de estos.

Figura 1: Primer fragmento del código

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
#objeto medalla
class Medallas:
    _name = ''
    _gold = ''
    _gold_range = ''
    _silver = ''
    _silver_range = ''
    _bronze = ''
    _bronze_range = ''

    def gt (self,other):
        if self._gold > other._gold:
            return True
        if self._gold < other._gold:
            return False
        if self._gold == other._gold:
            if self._silver > other._silver:
                return True
            if self._silver < other._silver:
                return False
            if self._silver == other._silver:
                if self._bronze > other._bronze:
                    return True
                return False

class predictor:
    def __init__(self):
        #leemos los datos
        file = pd.read_csv('panamericanos.csv')
        participantes = pd.read_csv('participantes.csv')
        #separamos los datos por países
        matrix = np.array(file)
        self.part_list = np.array(participantes)
        countries = {a : i for i, a in enumerate(set([a[0] for a in matrix[:,
1:2]]))}
        self.dic = dict(countries)
        for i in self.dic:
            self.dic[i] = []
        for i in matrix:
            self.dic[i[1]].append([i[0],[i[2]], [i[3]], [i[4]]])

    #calcula las medallas para un país específico
    def calculate_medals(self,current_data,current_target,year):
        pr = PolynomialFeatures(degree=2)
        pr4 = PolynomialFeatures(degree=4)
        poly_data = pr.fit_transform(current_data)
        poly_data4 = pr4.fit_transform(current_data)
        pr.fit(poly_data,current_target)
        pr4.fit(poly_data4,current_target)
        linear_reg = LinearRegression(normalize = True)
        linear_reg.fit(poly_data,current_target)
        linear_reg4 = LinearRegression(normalize = True)
        linear_reg4.fit(poly_data4,current_target)
        temp = linear_reg.predict(pr.fit_transform(year))
```

En este fragmento se puede apreciar como se importan las librerías anteriormente mencionadas, seguido de la implementación de la clase medalla para su posterior uso en el programa y se da paso a la implementación del algoritmo que resuelve nuestro problema.

Figura 2: Segundo fragmento del código

```
temp2 = linear_reg4.predict(pr4.fit_transform(year))
range = '['+str(self.remove_negatives(temp2[0][0]))+' - ' +
'+str(self.remove_negatives(temp[0][0]))+']'
return self.remove_negatives(temp[0][0])+self.remove_negatives(temp2[0][0])/2,range

def remove_negatives(self,temp):
    if (temp < 0):
        return 0
    return temp

def get_list(self,year):
    medallas2019 = []
    for item in self.dic:
        if item in self.part_list:
            newCountry = Medallas()
            newCountry._name = item
            value = self.dic[item][:]
            data = [x[0] for x in value]
            target = [x[1] for x in value]
            temp,range = self.calculate_medals(data,target,[[year]])
            newCountry._gold = np rint(self.remove_negatives(temp))
            newCountry._gold range=range
            target = [x[2] for x in value]
            temp,range = self.calculate_medals(data,target,[[year]])
            newCountry._silver = np rint(self.remove_negatives(temp))
            target = [x[3] for x in value]
            temp,range = self.calculate_medals(data,target,[[year]])
            newCountry._bronze = np rint(self.remove_negatives(temp))
            medallas2019.append(newCountry)
    medallas2019.sort(reverse = True)
    return medallas2019

def get_graphics(self,country):
    value = self.dic[country][:]
    current_data = [x[0] for x in value]
    for item in range(3):
        current_target = [x[item+1] for x in value]
        pr = PolynomialFeatures(degree=2)
        pr4 = PolynomialFeatures(degree=4)
        poly_data = pr.fit_transform(current_data)
        poly_data4 = pr4.fit_transform(current_data)
        pr.fit(poly_data,current_target)
        pr4.fit(poly_data4,current_target)
        linear_reg = LinearRegression(normalize = True)
        linear_reg.fit(poly_data,current_target)
        linear_reg4 = LinearRegression(normalize = True)
        linear_reg4.fit(poly_data4,current_target)
        plt.scatter(current_data,current_target, color = 'blue',
label="medallas historicas")

plt.plot(current_data,linear_reg.predict(pr.fit_transform(current_data)),
color="red",label="prediccion polinomio grado 2")

plt.plot(current_data,linear_reg4.predict(pr4.fit_transform(current_data)),
color="orange", label="prediccion polinomio grado 4")
plt.legend()
if(item == 0):
    plt.xlabel('Oro')
```

Figura 3: Continuación del segundo fragmento

```
        if(item == 1):
            plt.xlabel('Plata')
        if(item == 2):
            plt.xlabel('Bronce')
        plt.show()

a = predictor()
for item in a.get_list(2019):
    print(item._gold_range)
```

Con el código anterior dado los diferentes grados del polinomio se obtienen como resultados las siguientes tablas:

Figura 4: Resultado obtenido con el polinomio de grado 2

```
1 EstadosUnidos 70.0 70.0 71.0
2 Canada 60.0 52.0 49.0
3 Brasil 56.0 48.0 75.0
4 Cuba 38.0 19.0 35.0
5 Mexico 35.0 40.0 47.0
6 Argentina 33.0 37.0 44.0
7 Colombia 29.0 22.0 33.0
8 Venezuela 13.0 29.0 32.0
9 RepublicaDominicana 8.0 11.0 17.0
10 Ecuador 8.0 9.0 15.0
11 Chile 7.0 16.0 24.0
12 Guatemala 7.0 3.0 6.0
13 Jamaica 3.0 5.0 1.0
14 PuertoRico 3.0 2.0 10.0
15 Peru 2.0 4.0 9.0
16 TrinidadYTobago 2.0 3.0 2.0
17 Bahamas 2.0 2.0 2.0
18 SantaLucia 1.0 0.0 0.0
19 Uruguay 0.0 2.0 3.0
```

Figura 5: Resultado obtenido con el polinomio de grado 4

```
1 EstadosUnidos 89.0 51.0 56.0
2 Canada 79.0 68.0 75.0
3 Brasil 44.0 33.0 58.0
4 Colombia 38.0 14.0 37.0
5 Cuba 26.0 35.0 35.0
6 Mexico 26.0 34.0 59.0
7 Argentina 17.0 24.0 24.0
8 Ecuador 11.0 14.0 21.0
9 Guatemala 10.0 0.0 0.0
10 Chile 6.0 12.0 27.0
11 Peru 5.0 4.0 4.0
12 TrinidadYTobago 4.0 4.0 1.0
13 Venezuela 3.0 22.0 16.0
14 PuertoRico 2.0 2.0 15.0
15 Jamaica 1.0 4.0 0.0
16 Bahamas 1.0 2.0 2.0
17 CostaRica 1.0 1.0 1.0
18 SantaLucia 1.0 0.0 0.0
19 RepublicaDominicana 0.0 15.0 12.0
```

Figura 6: Resultado final producto de la unión de las predicciones anteriores

```
1 EstadosUnidos 79.0 60.0 64.0
2 Canada 70.0 60.0 62.0
3 Brasil 50.0 41.0 66.0
4 Colombia 33.0 18.0 35.0
5 Cuba 32.0 27.0 35.0
6 Mexico 30.0 37.0 53.0
7 Argentina 25.0 31.0 34.0
8 Ecuador 9.0 11.0 18.0
9 Venezuela 8.0 26.0 24.0
10 Guatemala 8.0 1.0 3.0
11 Chile 6.0 14.0 25.0
12 RepublicaDominicana 4.0 13.0 14.0
13 Peru 3.0 4.0 7.0
14 TrinidadYTobago 3.0 4.0 1.0
15 PuertoRico 3.0 2.0 13.0
16 Jamaica 2.0 4.0 [[0.]]
17 Bahamas 2.0 2.0 2.0
18 SantaLucia 1.0 0.0 0.0
19 Paraguay 0.0 1.0 3.0
```


3. Conclusiones

Como se pudo apreciar los resultados obtenidos por el programa se asemejan a la actuación real de cada uno de los países a los largo de la historia, salvo en excepciones donde se rompieron los pronósticos, señal de un buen funcionamiento y la aceptabilidad del pronóstico obtenido.

4. Anexos

Figura 7: Gráfica histórica de los oro de Cuba

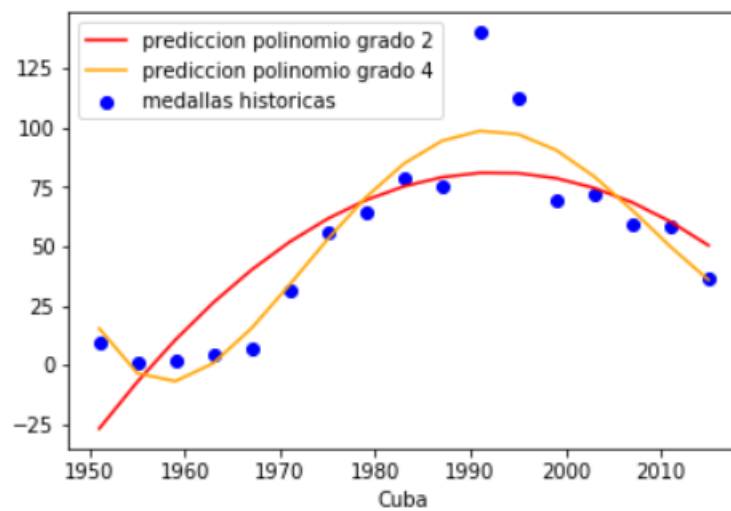


Figura 8: Gráfica histórica de los bronce de Cuba

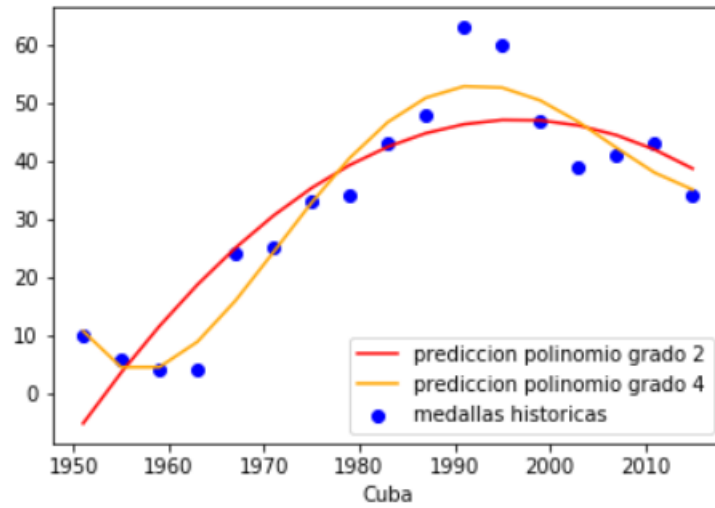


Figura 9: Gráfica histórica de las plata de Cuba

