Proyecto final

Brandon Francisco Hernández 154328
Emilia Sofia Spinola 172103
Miguel Ángel Cifuentes Jimenez 168274
Francesca Perrone 179682

Fundamentos Teóricos

El propósito del algoritmo de aproximación es expresar el comportamiento de una variable como una función de una serie de variables independiente

$$y = f(v_1, v_2, \ldots, v_n) = f(\hat{v})$$

En esta ecuación ***y*** es una variable dependiente y ***v*** funciona como un set de variables independientes que expresan a la variable y en la función f.

La aproximación:

$$y = c_1X_1 + c_2X_2 + \ldots + c_mX_m$$

Aquí *c* representa a los coeficientes, *m* representa el número de coeficientes que se desean y *Xi* denota las combinaciones de variables independientes $(X_i = f_i(\hat{v}))$. Dependiendo de la forma en como estas combinaciones se definan, se pueden obtener diferentes aproximaciones.

El objetivo del algoritmo de aproximación (Exchange Algorithm) es encontrar los valores de los coeficientes en nuestra función, de tal forma que los valores aproximados logren minimizar la diferencia entre los valores conocidos de la variable dependiente f y los valores calculados de todos los objetos en la muestra. Para determinar los coeficientes de aproximación necesitamos definir un error.

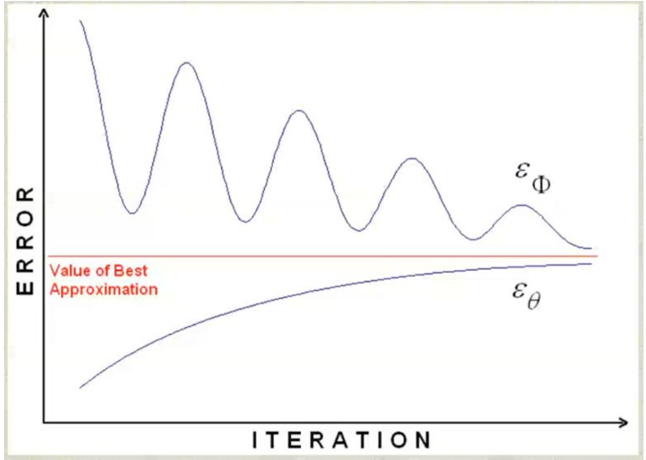
$$\varepsilon = max(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m) \text{ donde } \varepsilon_1 = |f_i - y_i|$$

El ajuste mínimo, el ajuste que tiene el error más pequeño, es aquel en el cual los errores de aproximación son todos del mismo tamaño absoluto (todos los errores dan un valor absoluto igual entre sí), y esto se consigue por medio de la aproximación minimax (el mínimo de los errores máximos). Recordemos que esto se obtiene como resultado de encontrar el error utilizando la regla de cramer. Para obtener el signo del error utilizamos el teorema que nos enseñó el profesor en clase, que dice que si los cofactores de una columna de un determinante se multiplican por los elementos de una columna distinta y se suman el resultado debe ser cero.

Ahora hablaremos del algoritmo de ascenso, el cual se utiliza para encontrar los dos errores que nos interesan. El primer error que nos interesa se obtiene del valor absoluto del error aproximado más grande del set interno. Este error se denomina ε_θ. El segundo error que nos interesa se obtiene del valor absoluto del error aproximado más grande del set externo, a este se le denomina ε_Φ.

Ambos errores se aproximan entre sí hasta alcanzar la condición de convergencia teórica:

$$\varepsilon_\theta(t) \geq \varepsilon_\Phi(t)$$



Si logramos encontrar la mejor aproximación para el set interno, por definición estamos encontrando el error más pequeño en ese set, y la aproximación polinomial correspondiente de la iteración t (es decir yt(X)) es la cual tiene los coeficientes que minimizan la diferencia entre el set de datos observados y la nuestra función 1 en t.

Si los valores que obtenemos de yt(X) para los objetos en el set externo nos dan un valor de error igual o menor que el error del set interno (condición de convergencia) , entonces podemos decir que yt(X) es la mejor solución general.

Si, los errores que obtenemos de los valores de yt(X) en el set externo son mayores que los valores que obtenemos de los yt(X) del set interno, entonces tenemos que intercambiar un objeto interno con un objeto externo y repetir todo el proceso.

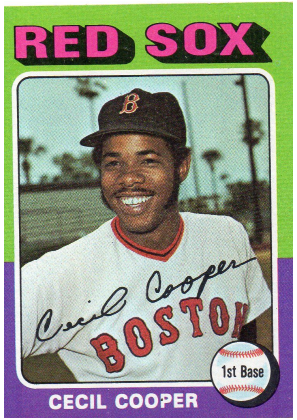
Análisis metodológico

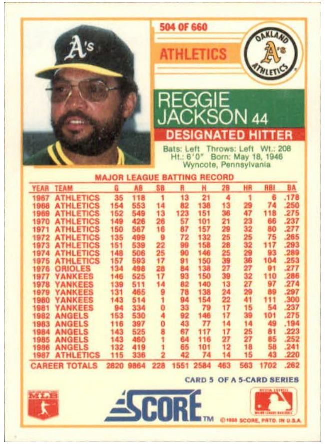
Objetivo: determinar si una persona es propensa a contagiarse de COVID-19 dadas algunas condiciones de salud.

Uso de dos conjuntos de datos.

- Fase de Entrenamiento
- Fase de Prueba

Aplicaciones





504 OF 660

ATHLETICS

REGGIE
JACKSON 44

DESIGNATED HITTER

Bats: Left Throws: Left Wt.: 208
Ht.: 6'0" Born: May 18, 1946
Wyncote, Pennsylvania

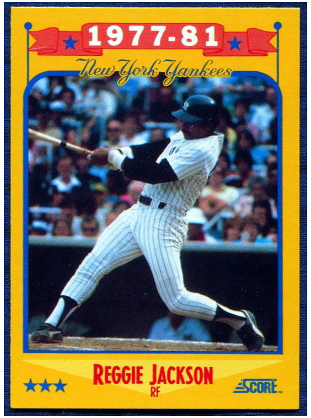
MAJOR LEAGUE BATTING RECORD

YEAR	TEAM	G	AB	SB	R	H	2B	HR	RBI	BA
1967	ATHLETICS	35	118	1	13	21	4	1	6	.178
1968	ATHLETICS	154	553	14	82	138	13	29	74	.250
1969	ATHLETICS	152	549	13	123	151	36	47	118	.275
1970	ATHLETICS	149	426	26	57	101	21	23	66	.237
1971	ATHLETICS	150	567	16	87	157	29	32	80	.277
1972	ATHLETICS	135	499	9	72	132	25	25	75	.265
1973	ATHLETICS	151	539	22	99	158	28	32	117	.293
1974	ATHLETICS	148	506	25	90	146	25	29	93	.289
1975	ATHLETICS	157	593	17	91	150	39	36	104	.253
1976	ORIOLES	134	498	28	84	138	27	27	91	.277
1977	YANKEES	146	525	17	93	150	39	32	110	.286
1978	YANKEES	139	511	14	82	140	13	27	97	.274
1979	YANKEES	131	465	9	78	138	24	29	89	.297
1980	YANKEES	143	514	1	94	154	22	41	111	.300
1981	YANKEES	94	334	0	33	79	17	15	54	.237
1982	ANGELS	153	530	4	92	146	17	39	101	.275
1983	ANGELS	116	397	0	43	77	14	14	49	.194
1984	ANGELS	143	525	8	67	117	17	25	81	.223
1985	ANGELS	143	460	1	64	116	27	27	85	.252
1986	ANGELS	132	419	1	65	101	12	18	58	.241
1987	ATHLETICS	115	336	2	42	74	14	15	43	.220
CAREER TOTALS		2820	9864	228	1551	2584	463	563	1702	.262

CARD 5 OF A 5-CARD SERIES

SCORE™

©1988 SCORE, PRTD. IN U.S.A.



Programación

- Se generó los conjuntos de entrenamiento y prueba
- Optimización del código para diferentes grados máximos de cada variable
- Almacenamiento de los errores de los conjuntos externos
- Almacenamiento de todas las combinaciones de coeficientes y potencias
- Cálculo de resultados para datos de prueba
- Comparación entre resultados y datos originales
- Generación de archivo para guardar los coeficientes

Resultados

RMSE

```
In [89]: print(min(listaRMSE))
0.39916021020675224
```

c[00000]=-0.24933471010490585
c[00001]=2.2078189365668357
c[00010]=23.00121174644182
c[00011]=-23.874846929377956
c[00100]=-0.6976080850953775
c[00101]=-2.0496470441254204
c[00110]=-21.80382834716506
c[00111]=23.716328569920435
c[01000]=-0.0011269876832046752
c[01001]=-2.737840023849679
c[01010]=-46.50327741536117
c[01011]=47.30224429620281
c[01100]=1.4110771749621212
c[01101]=2.5509848595574995
c[01110]=45.09256004647704
c[01111]=-47.11498409357338
c[10000]=-0.0010291890815343349
c[10001]=-1.7072603537601827
c[10010]=3.3119570792047477
c[10011]=-2.437516885538301
c[10100]=10.455042950881095
c[10101]=-7.472189859405564
c[10110]=-14.267676884522333
c[10111]=12.118473754312012
c[11000]=0.0021469229748506246
c[11001]=2.736223556873169
c[11010]=-10.064786551375235
c[11011]=9.265133992334404
c[11100]=-13.04546519726694
c[11101]=8.171394450260474
c[11110]=23.11295392978464
c[11111]=-20.177038668410695