

CAPITULO IV PRONÓSTICOS

4.1 Introducción.

La mayoría de la actividades en los diferentes empresas públicas y privadas necesitan información futura, esto es la predicción del comportamiento de los datos en series de tiempo, ya que mucha de la información tiene variación estacional o cíclica de la tendencia donde los modelos de pronóstico nos permite hallar el valor esperado de acuerdo al comportamiento de los datos y permita a quien toma las decisiones poder realizar cambios en el comportamiento de los sistemas como:

Patrones de demanda

Calculo de la temperatura, humedad, viento

Compras ventas por periodos

Proyección de utilidades perdidas en meses años

Cartera de inversiones, variaciones de la bolsa de valores

Modelos de pronóstico

Promedio Móvil

Alisamiento exponencial(suavización exponencial)

Regresión lineal

Regresión cuadrática

4.2 Promedio Móvil.

El método de promedio móvil se utiliza cuando se requiere dar más importancia a los datos recientes, el pronóstico se lo calcula con el promedio de n valores de los datos más recientes de la siguiente forma:

$$\text{Promedio Móvil} = \left(\sum_{i=1}^n [(n \text{ puntos de datos más recientes})] \right) / n$$

Ejemplo: Para este ejemplo vamos a utilizar una serie cronológica de las exportaciones de Ecuador del período 2003-2017, datos tomados del sitio estadístico del BANCO CENTRAL DEL ECUADOR. El análisis de pronósticos está enfocado a procesar los datos con los tres métodos promedio móvil, suavización exponencial y por ultimo con la regresión de los mínimos cuadrados.

Al final realizaremos un análisis de los datos con las diferentes opciones para realizar pronósticos.

Table: **Tabla de Exportaciones Año 2003-2017**

Años	EXPORTACIONES (MILLONES \$)
2003	5,501.0
2004	6,232.7
2005	8,118.3

Años	EXPORTACIONES (MILLONES \$)
<i>2006</i>	10,137.0
<i>2007</i>	10,449.5
<i>2008</i>	12,794.6
<i>2009</i>	9,939.1
<i>2010</i>	13,193.0
<i>2011</i>	16,036.2
<i>2012</i>	18,496.9
<i>2013</i>	18,709.3
<i>2014</i>	19,363.5
<i>2015</i>	16,521.5
<i>2016</i>	15,175.4
<i>2017</i>	16,927.0

Fuente: Banco Central del Ecuador

In [2]:

```

1  ### Promedio Móvil
2
3  # Vamos a crear un DataFrame con Los datos y luego procederemos a calcular el promedio
4
5  import pandas as pd
6  import numpy as np
7  import matplotlib.pyplot as plt
8  # el DataFrame se llama movil
9  exporta = {'Año':[2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018],
10             'Exportaciones':[5501.0, 6232.7, 8118.3, 10137.00, 10449.50, 12794.60, 9939.10, 13193.00, 16036.2, 18496.90, 18709.30, 19363.50, 16927.00]}
11
12
13  movil = pd.DataFrame(exports)
14  # mostramos los 5 primeros registros
15  movil.head()
16
17  # calculamos para la primera media móvil MMO_3
18
19  for i in range(0,movil.shape[0]-2):
20      movil.loc[movil.index[i+2],'MMO_3'] = np.round(((movil.iloc[i,1]+movil.iloc[i+1,1]+movil.iloc[i+2,1])/3))
21
22  # calculamos para la segunda media móvil MMO_4
23  for i in range(0,movil.shape[0]-3):
24      movil.loc[movil.index[i+3],'MMO_4'] = np.round(((movil.iloc[i,1]+movil.iloc[i+1,1]+movil.iloc[i+2,1]+movil.iloc[i+3,1])/4))
25
26  # calculamos la proyección final
27
28  proyeccion = movil.iloc[12:,[1,2,3]]
29  p1,p2,p3 =proyeccion.mean()
30  # incorporamos al DataFrame
31  a = movil.append({'Año':2018,'Exportaciones':p1, 'MMO_3':p2, 'MMO_4':p3},ignore_index=True)
32  # mostramos los resultados
33
34  a['e_MM3'] = a['Exportaciones']-a['MMO_3']
35  a['e_MM4'] = a['Exportaciones']-a['MMO_4']
36  a
37
38

```

Out[2]:

	Año	Exportaciones	MMO_3	MMO_4	e_MM3	e_MM4
0	2003.0	5501.000000	NaN	NaN	NaN	NaN
1	2004.0	6232.700000	NaN	NaN	NaN	NaN
2	2005.0	8118.300000	6617.300000	NaN	1501.0	NaN
3	2006.0	10137.000000	8162.700000	7497.200000	1974.3	2639.8
4	2007.0	10449.500000	9568.300000	8734.400000	881.2	1715.1
5	2008.0	12794.600000	11127.000000	10374.800000	1667.6	2419.8
6	2009.0	9939.100000	11061.100000	10830.000000	-1122.0	-890.9
7	2010.0	13193.000000	11975.600000	11594.000000	1217.4	1599.0
8	2011.0	16036.200000	13056.100000	12990.700000	2980.1	3045.5
9	2012.0	18496.900000	15908.700000	14416.300000	2588.2	4080.6

	Año	Exportaciones	MMO_3	MMO_4	e_MM3	e_MM4
10	2013.0	18709.300000	17747.500000	16608.900000	961.8	2100.4
11	2014.0	19363.500000	18856.600000	18151.500000	506.9	1212.0
12	2015.0	16521.500000	18198.100000	18272.800000	-1676.6	-1751.3
13	2016.0	15175.400000	17020.100000	17442.400000	-1844.7	-2267.0
14	2017.0	16927.000000	16208.000000	16996.800000	719.0	-69.8
15	2018.0	16207.966667	17142.066667	17570.666667	-934.1	-1362.7

La fila 15 del DataFrame contiene los pronósticos del promedio móvil MMO_3, MMO_4, y los errores e_MM3, e_MM4

luego calculamos el promedio de cada una de las columnas y

```
In [3]: 1 # calculamos el promedio de los cada una de las columnas de df
2 m1,m2,m3,m4,m5,m6 =a.mean()
3 print('Error Media Móvil 3 = ',round(m5),'Error Media Móvil 4 = ',round(m6))
4
5
```

Error Media Móvil 3 = 673 Error Media Móvil 4 = 959

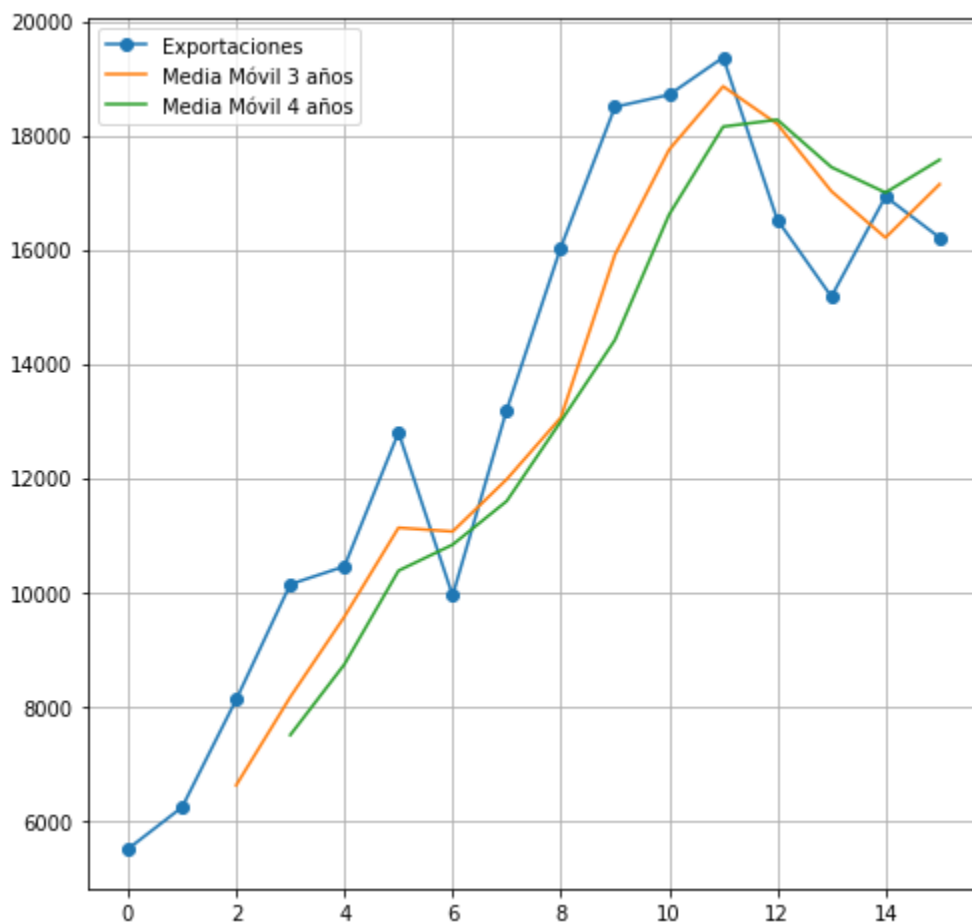
Graficando los resultados

```

In [4]: 1 %%matplotlib inline
2
3 plt.figure(figsize=[8,8])
4 plt.grid(True)
5 plt.plot(a['Exportaciones'],label='Exportaciones',marker='o')
6 plt.plot(a['MMO_3'],label='Media Móvil 3 años')
7 plt.plot(a['MMO_4'],label='Media Móvil 4 años')
8 plt.legend(loc=2)
9
10

```

Out[4]: <matplotlib.legend.Legend at 0x255407109e8>



4.3 Suavización Exponencial

"La suavización exponencial utiliza un promedio ponderado de valores de series de tiempo pasadas como pronóstico; es un caso especial del método de promedios móviles ponderados en el cual seleccionamos sólo un peso, el peso para la observación más reciente. Los pesos para los demás valores se calculan de forma automática y se vuelven cada vez más pequeños a medida que las observaciones se alejan en el pasado.

-Anderson, 11ed

El modelo de suavización exponencial básico es:

$$S_n = \alpha X_{n-1} + (1-\alpha)S_{n-1}$$

- S_n = Estimación Alisada. Es una Combinación Lineal de todos las demandas pasadas con coeficientes que disminuyen en forma geométrica de acuerdo con la edad de observación.
- X_{n-1} = valor real de la serie de tiempo en el periodo n-1
- S_{n-1} = pronóstico de la serie de tiempo del periodo n-1
- α = constante de suavización ($0 \leq \alpha \leq 1$)

Alfa 0.1

table:

Año	Exportaciones	Estimación Alisada	Error
n	X_n	$S_n = \alpha X_{n-1} + (1-\alpha)S_{n-1}$	$Er = X_n - S_n$
2003	5,501.0	0	0
2004	6,232.7	5,501.00	731.70
2005	8,118.3	5,574.17	2544.13

In [5]:

```
1  ### Alisamiento Exponencial
2
3  # Vamos a crear un DataFrame con Los datos y luego procederemos a calcular el promedio
4
5  import pandas as pd
6  import numpy as np
7  import matplotlib.pyplot as plt
8  # el DataFrame se llama movil
9  exporta = {'Año':[2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018],
10             'Exportaciones':[5501.0, 6232.7, 8118.3, 10137.00, 10449.50, 12794.6, 9939.1, 13193.00, 16036.2, 18496.90, 18709.30, 19363.50, 16521.5, 15175.4, 16927.00]}
11
12
13  movil = pd.DataFrame(exporta)
14  # mostramos los 5 primeros registros
15  movil.head()
16  alfa = 0.1
17  unoalfa = 1. - alfa
18
19  for i in range(0,movil.shape[0]-1):
20      movil.loc[movil.index[i+1], 'SN'] = np.round(movil.iloc[i,1],1)
21
22  for i in range(2,movil.shape[0]):
23      movil.loc[movil.index[i], 'SN'] = np.round(movil.iloc[i-1,1],1)*alfa + np.round(movil.iloc[i-1,2],1)*unoalfa
24  i=i+1
25  p1=0
26  p2=np.round(movil.iloc[i-1,1],1)*alfa + np.round(movil.iloc[i-1,2],1)*unoalfa
27  a = movil.append({'Año':2018, 'Exportaciones':p1, 'SN':p2},ignore_index=True)
28  a
29
30  # movil
31
```

Out[5]:

	Año	Exportaciones	SN
0	2003.0	5501.0	NaN
1	2004.0	6232.7	5501.00
2	2005.0	8118.3	5574.17
3	2006.0	10137.0	5828.61
4	2007.0	10449.5	6259.44
5	2008.0	12794.6	6678.41
6	2009.0	9939.1	7290.02
7	2010.0	13193.0	7554.91
8	2011.0	16036.2	8118.71
9	2012.0	18496.9	8910.45
10	2013.0	18709.3	9869.05
11	2014.0	19363.5	10753.03
12	2015.0	16521.5	11614.05
13	2016.0	15175.4	12104.84
14	2017.0	16927.0	12411.86
15	2018.0	0.0	12863.41

4.4 Regresión Lineal y cuadrada

In [7]:

```
1  # Importar libreria numpy
2  import numpy as np
3
4  # datos experimentales
5  # el DataFrame se llama movil
6
7  exporta = {'Año':[2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020],
8             'Exportaciones':[5501.0, 6232.7, 8118.3, 10137.00, 10449.50, 12794.00, 13193.00, 16036.2, 18496.90, 18709.30, 19363.50, 16927.00]}
9
10
11 a = pd.DataFrame(exporta)
12 x = a.index.values
13 y= a["Exportaciones"]
14 # ajuste de la recta (polinomio de grado 1  $f(x) = ax + b$ )
15 p = np.polyfit(x,y,1) # 1 para lineal, 2 para polinomio ...
16 p0,p1 = p
17 print ("El valor de p0 = ", p0, "Valor de p1 = ", p1)
18
```

El valor de p0 = 910.1474999999998 Valor de p1 = 6801.967499999998


```

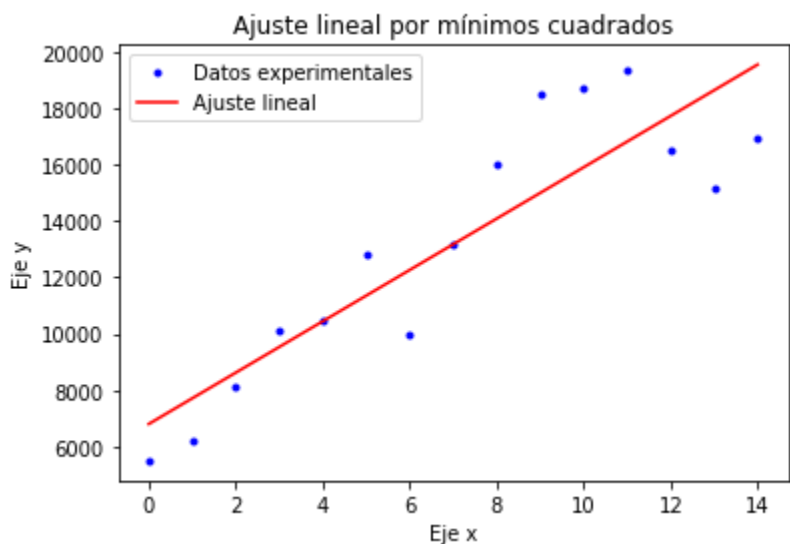
In [7]: 1 from matplotlib import pyplot as plt
2
3 #  $y(x) = poX + p1 = 49.53676471X - 242.41911765$ 
4
5 # calculamos los valores ajustados y_ajuste
6
7 y_ajuste = p[0]*x + p[1]
8
9 print (y_ajuste)
10 # dibujamos los datos experimentales de la recta
11
12 p_datos = plt.plot(x,y, 'b. ')
13 # Dibujamos la recta de ajuste
14
15 p_ajuste = plt.plot(x,y_ajuste, 'r-')
16
17 plt.title('Ajuste lineal por mínimos cuadrados')
18
19 plt.xlabel('Eje x')
20 plt.ylabel('Eje y')
21
22 plt.legend(('Datos experimentales', 'Ajuste lineal',), loc="upper left")
23 plt.show()
24

```

```

[ 6801.9675  7712.115   8622.2625  9532.41   10442.5575  11352.705
 12262.8525 13173.      14083.1475 14993.295   15903.4425 16813.59
 17723.7375 18633.885   19544.0325]

```



Regresión Lineal cuadrada

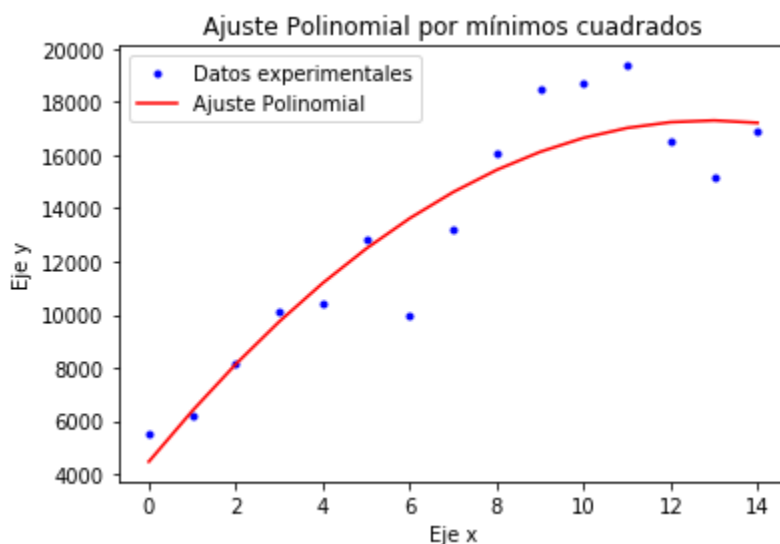
$$f(x) = ax^2 + bx + c$$

```
In [8]: 1 p = np.polyfit(x,y,2)
2 p1,p2,p3 = p
3
4 print ("El valor de p0 = ", p0, "Valor de p1 = ", p1, " el valor de p2 = ",p2)
5
6
7
```

El valor de p0 = 910.1474999999998 Valor de p1 = -76.89145927601825 el valor de p2 = 1986.6279298642573

$$F(x) = p_0x^2 + p_1x + p_2$$

```
In [9]: 1 from matplotlib import pyplot as plt
2
3 # calculamos los valores ajustados y_ajuste
4
5 y_ajuste = p[0]*x*x + p[1]*x + p[2]
6
7 # dibujamos los datos experimentales de la recta
8
9 p_datos =plt.plot(x,y,'b.')
10
11 # Dibujamos la curva de ajuste
12
13 p_ajuste = plt.plot(x,y_ajuste, 'r-')
14
15 plt.title('Ajuste Polinomial por mínimos cuadrados')
16
17 plt.xlabel('Eje x')
18 plt.ylabel('Eje y')
19
20 plt.legend(('Datos experimentales','Ajuste Polinomial'),, loc="upper left")
21 plt.show()
22
23
```



In [10]:

```
1 x
2 #x.size
3
```

Out[10]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14],
dtype=int64)

In [11]:

```
1 # 3 proyecciones de los siguientes días
2 n=x.size
3 x1 = []
4 x2 = []
5
6
7 for i in [15,16,17]:
8     y1_ajuste = p[0]*i*i + p[1]*i + p[2]
9     print (f" z = {i} w = {y1_ajuste}")
10    x1.append(i)
11    x2.append(y1_ajuste)
12
13 a["y_ajuste"]=y_ajuste
14
15 dp = pd.DataFrame({'Año':[2018,2019,2020], 'Exportaciones':[0,0,0], 'y_ajuste':x2})
16 dp
17 a = a.append(dp,ignore_index=True)
18 a
19
```

```
z = 15 w = 16968.43384615386
z = 16 w = 16571.426538461554
z = 17 w = 16020.636312217212
```

Out[11]:

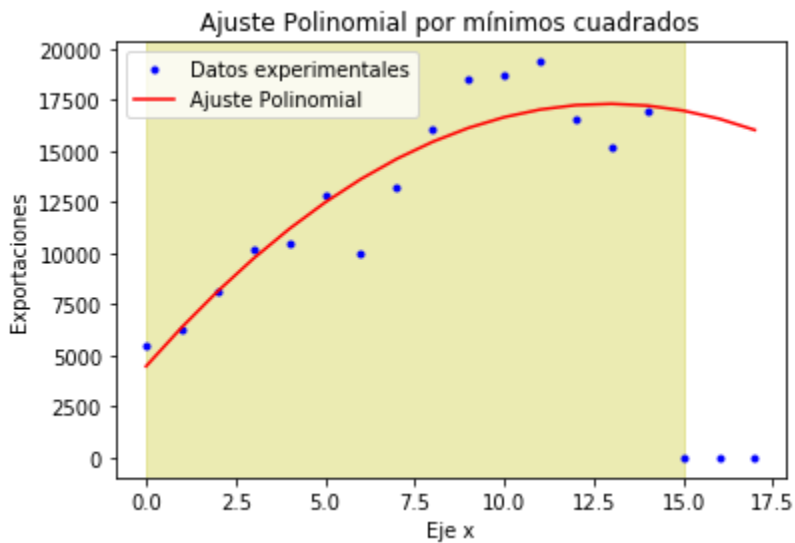
	Año	Exportaciones	y_ajuste
0	2003	5501.0	4469.593235
1	2004	6232.7	6379.329706
2	2005	8118.3	8135.283258
3	2006	10137.0	9737.453891
4	2007	10449.5	11185.841606
5	2008	12794.6	12480.446403
6	2009	9939.1	13621.268281
7	2010	13193.0	14608.307240
8	2011	16036.2	15441.563281
9	2012	18496.9	16121.036403
10	2013	18709.3	16646.726606
11	2014	19363.5	17018.633891
12	2015	16521.5	17236.758258
13	2016	15175.4	17301.099706
14	2017	16927.0	17211.658235
15	2018	0.0	16968.433846
16	2019	0.0	16571.426538
17	2020	0.0	16020.636312

Graficamos el pronóstico

Ya tenemos el DataFrame con los pronósticos incluidos ahora si vamos a visualizarlos tomamos los datos del DataFrme

In [12]:

```
1 x = a.index.values
2 y_ajuste = a["y_ajuste"]
3 y= a["Exportaciones"]
4
5 p_datos =plt.plot(y,'b.')
6 p_ajuste = plt.plot(x,y_ajuste, 'r-')
7
8 plt.title('Ajuste Polinomial por mínimos cuadrados')
9
10 plt.xlabel('Eje x')
11 plt.ylabel('Exportaciones')
12 plt.axvspan(0,15,alpha=0.3,color='y') # encajonamos los datos iniciales
13 plt.legend(('Datos experimentales','Ajuste Polinomial'),, loc="upper left")
14 plt.show()
15
```



In [13]:

```
1 # Aquí en la gráfica ya observamos los nuevos valores de la proyección
2
```

In [14]:

```
1 # se pueden observar los valores de los pronósticos de los tres períodos siguientes:
2
3 dp
4
```

Out[14]:

	Año	Exportaciones	y_ajuste
0	2018	0	16968.433846
1	2019	0	16571.426538
2	2020	0	16020.636312

Conclusiones

como podemos observar con los tres modelos de pronóstico revisados, vemos que el de regresión lineal o polinómica puede ser más efectivo en cualquier trabajo de simulación, es más éste método es uno de los primeros que se utiliza para análisis de datos con Data Science, ya que con la función $f(x) = ax^2 + bx + c$ nos facilita realizar la comprobación con cualquier valor de x, y no solo de un siguiente pronóstico:

In []:

1