

## BOLETÍN DE SIMULACION

ESTUDIANTE: ANGEL RUIZ

### Introducción

[1] El producto interior bruto (PIB) es un indicador económico que refleja el valor monetario de todos los bienes y servicios finales producidos por un país o región en un determinado periodo de tiempo, normalmente un año. Se utiliza para medir la riqueza que genera un país. También se conoce como producto bruto interno (PBI). Por otra parte la regresión es un proceso estadístico para estimar las relaciones entre variables. Incluye muchas técnicas para el modelado y análisis de diversas variables, logrando así con esta técnica poder predecir futuros valores del PIB en nuestro país con una alta precisión

### Importación de librerías

A continuación se importaran todas las librerías que se utilizaran en el proceso del análisis y simulación de los datos.

```
In [19]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime,timedelta
from sklearn.model_selection import train_test_split
import datetime as dt
from sklearn.linear_model import LinearRegression
import altair as alt
import altair_viewer
from sklearn.preprocessing import PolynomialFeatures
```

### Importación de datos

Se importa el dataset que se utilizará para obtener todos los datos necesarios

```
In [2]: df = pd.read_csv('./Datos.csv',encoding='utf-8')
df = df.fillna(0)
df.head()
```

|   | Country Name | Country Code | Indicator Name                                  | Indicator Code | 1960     | 1961     | 1962     | 1963     | 1964     | 1965      | ... | 2010      | 2011      | 2012      | 2013      | 2014      | 2015      | 2016      | 2017      | 2018      |
|---|--------------|--------------|---|----------------|----------|----------|----------|----------|----------|-----------|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0 | Aruba        | ABW          | Exportaciones de bienes y servicios (% del PIB) | NE.EXP.GNFS.ZS | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000  | ... | 61.042300 | 67.813322 | 65.241349 | 76.509512 | 77.555556 | 73.517030 | 71.294029 | 73.332115 | 0.000000  |
| 1 | Afganistán   | AFG          | Exportaciones de bienes y servicios (% del PIB) | NE.EXP.GNFS.ZS | 4.132233 | 4.453443 | 4.878051 | 9.171601 | 8.888893 | 11.258279 | ... | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  |
| 2 | Angola       | AGO          | Exportaciones de bienes y servicios (% del PIB) | NE.EXP.GNFS.ZS | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000  | ... | 61.543113 | 60.669948 | 55.940131 | 50.747084 | 44.695031 | 29.754599 | 28.124485 | 29.004100 | 40.836290 |
| 3 | Albania      | ALB          | Exportaciones de bienes y servicios (% del PIB) | NE.EXP.GNFS.ZS | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000  | ... | 27.979449 | 29.241501 | 28.937487 | 28.916354 | 28.212979 | 27.267393 | 28.977864 | 31.569821 | 31.591574 |
| 4 | Andorra      | AND          | Exportaciones de bienes y servicios (% del PIB) | NE.EXP.GNFS.ZS | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000  | ... | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  |

5 rows × 64 columns

### Preprocesamiento

Para una fácil lectura de los datos, intercambiamos las filas por columnasdata.describe().round(3)

```
In [7]: data = df.T
data.columns = df['Country Name']
data = data[4:]
data.head()
```

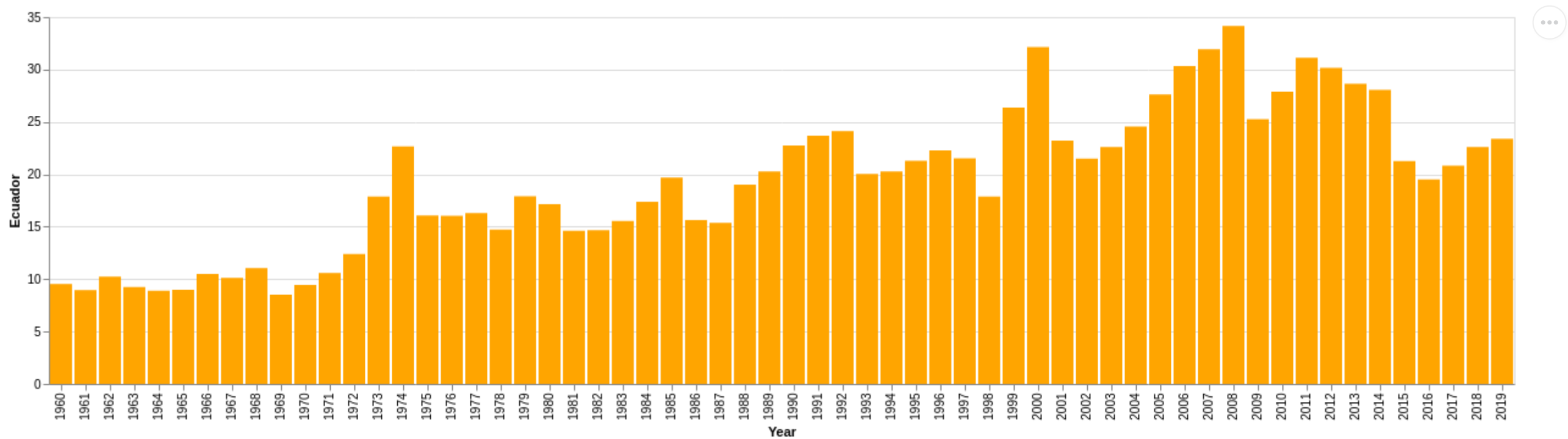
| Country Name | Aruba | Afganistán | Angola | Albania | Andorra | El mundo árabe | Emiratos Árabes Unidos | Argentina | Armenia | Samoa Americana | ... | Islas Vírgenes (EE.UU.) | Viet Nam | Vanuatu | Mundo | Samoa | Kosovo | Yemen, Rep. del | Sudáfrica | Zambia | Zimbabwe |
|--------------|-------|------------|--------|---------|---------|----------------|------------------------|-----------|---------|-----------------|-----|-------------------------|----------|---------|-------|-------|--------|-----------------|-----------|--------|----------|
| 1960         | 0     | 4.13223    | 0      | 0       | 0       | 0              | 0                      | 7.60405   | 0       | 0               | ... | 0                       | 0        | 0       | 0     | 0     | 0      | 0               | 29.5509   | 0      | 0        |
| 1961         | 0     | 4.45344    | 0      | 0       | 0       | 0              | 0                      | 5.99495   | 0       | 0               | ... | 0                       | 0        | 0       | 0     | 0     | 0      | 0               | 29.324    | 0      | 0        |
| 1962         | 0     | 4.87805    | 0      | 0       | 0       | 0              | 0                      | 4.69184   | 0       | 0               | ... | 0                       | 0        | 0       | 0     | 0     | 0      | 0               | 29.4069   | 0      | 0        |
| 1963         | 0     | 9.1716     | 0      | 0       | 0       | 0              | 0                      | 7.89045   | 0       | 0               | ... | 0                       | 0        | 0       | 0     | 0     | 0      | 0               | 28.6139   | 0      | 0        |
| 1964         | 0     | 8.88889    | 0      | 0       | 0       | 0              | 0                      | 5.56372   | 0       | 0               | ... | 0                       | 0        | 0       | 0     | 0     | 0      | 0               | 27.4359   | 0      | 0        |

5 rows × 264 columns

### Modelo Matemático

Para el modelo matemático utilizaremos una regresión lineal, el cual nos permitirá predecir valores del PIB en futuros años, para ello utilizaremos como entrada el año y como salida el PIB

```
In [13]: alt.Chart(data).mark_bar(color='orange').encode(
x=alt.X('Year',sort=alt.EncodingSortField(field="id", op="count", order='ascending')),
y='Ecuador')
).interactive()
```



### Dividimos los datos que utilizaremos en el modelo matemático

```
In [15]: X=data['Year'].values.reshape(-1,1)
y=data['Ecuador']
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=.2, random_state=42)
```

### Definimos nuestro modelo matematico, y lo entrenamos

```
In [16]: linear_regressor = LinearRegression()
linear_regressor.fit(X_train, y_train)
```

Out[16]: LinearRegression()

```
In [17]: print('Precisión: ', linear_regressor.score(X_train, y_train))
```

Precisión: 0.699017421534448

### La precisión de nuestro es muy baja, por ende definiremos un modelo polinomial

```
In [20]: poly_reg1 = PolynomialFeatures(degree = 4)
poly_reg = PolynomialFeatures(degree = 6)
X_poly = poly_reg1.fit_transform(X_train)
lin_reg_1 = LinearRegression()
lin_reg_1.fit(X_poly, y_train)
```

Out[20]: LinearRegression()

```
In [21]: print('Precisión: ', lin_reg_1.score(X_poly, y_train))
```

Precisión: 0.7692586951258074

### Una vez mejorada nuestra precisión, podemos predecir futuros valores del PIB en nuestro país

```
In [33]: X_poly2 = poly_reg1.fit_transform(X_test)
predi=lin_reg_1.predict(X_poly2)
print(predi)
```

[ 9.27911261 10.11672713 23.36966476 25.67232843 12.81258091 25.20719488  
22.18648424 25.88211406 12.40538785 24.23588445 25.77853391 25.82995263]

```
In [31]: print(y_test)
```

1960 9.54757  
1965 8.97966  
1996 22.2733  
2005 27.6182  
1973 17.878  
2014 28.0519  
1993 20.0358  
2008 34.1636  
1972 12.4079  
1917 20.8328  
2006 30.3331  
2010 27.895  
Name: Ecuador, dtype: object

### Conclusión

Podemos observar que nuestro modelo matemático tiene una precisión del 77% que es un valor aceptable para poder hacer uso del mismo, sin embargo con una refinación adecuada de los datos se podría lograr una mayor precisión y por ende predicciones aún más cercanas a la realidad. Al hacer uso de estas técnicas nos permite visualizar el futuro y poder tomar medidas con la prevención de problemas, o todo lo contrario tomar provecho una vez sabemos los posibles valores del PIB en próximos años.

### Referencias

[1] A. Sevilla Arias. Producto interior bruto (PIB). Economipedia. <https://economipedia.com/definiciones/producto-interior-bruto-pib.html>