



## Simulación

Tema: Simulación de Eventos.

Estudiante: Paulo Gonzalez

### Objetivo:

- Consolidar los conocimientos adquiridos en clase para desarrollar simulaciones.

### Enunciado:

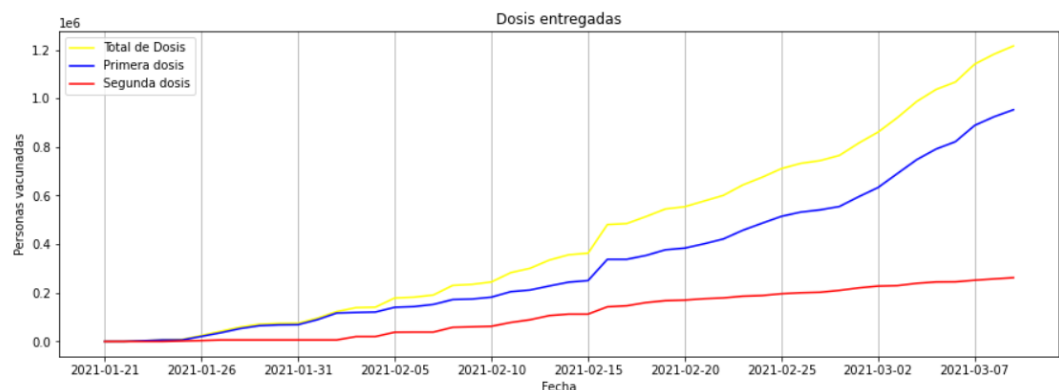
- Diseñe y desarrolle un modelo y/o *script* que permita simular el siguiente caso real:
  - Se tiene los datos del ecuador ([https://github.com/andrab/ecuacovid/tree/master/datos\\_crudos](https://github.com/andrab/ecuacovid/tree/master/datos_crudos)). En base a ello obtener los siguientes modelos:
    - Generar graficas para entender y procesar los datos:

- Generar gráficas y reportes del total de personas vacunadas.

Como primer punto hacemos la lectura de datos para poder generar los gráficos, además que nosotros damos una fecha inicial y hasta cuantos días después, para un mejor análisis se tomó 48 días después de la primera fecha de vacunación

```
: import matplotlib.dates as mtdt
import datetime as dt
import matplotlib.pyplot as plt
import pandas as pd
from datetime import datetime, timedelta
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
vacunados=pd.read_csv('vacunas.csv')
planVacunarse=pd.read_csv('vacunas_Ecuador.csv')
fabricantes=pd.read_csv('fabricantes.csv')
```

Una vez leído el dataset de las vacunas obtenemos este resultado desde el 21/01/2021 que fue la fecha en la que el Ecuador tuvo la vacuna para los médicos, aunque no fueron vacunados desde ese día; 48 días después este es el grafico y datos resultantes:





## Simulación

Tema: Simulación de Eventos.

Estudiante: Paulo Gonzalez

	fecha	dosis_total	primera_dosis	segunda_dosis
0	21/01/2021	0	0	0
1	22/01/2021	108	108	0
2	27/01/2021	2982	2982	0
3	04/02/2021	6228	6228	0
4	17/02/2021	8190	6228	1962

- **Generar grafico de pie por fabricante de la vacuna.**

Como punto de partida obtenemos la cantidad vacunas generadas de cada de uno los fabricantes, el nombre de las vacunas y su fecha de arribo

```
fabricantes=pd.read_csv('fabricantes.csv')
fabricantes=fabricantes.head(5)
fabricantes
```

Obteniendo como resultado la siguiente tabla:

	vaccine	total	arrived_at
0	Pfizer/BioNTech	8190	20/01/2021
1	Pfizer/BioNTech	16380	17/02/2021
2	Pfizer/BioNTech	17550	24/02/2021
3	Pfizer/BioNTech	31590	03/03/2021
4	Sinovac	20000	06/03/2021

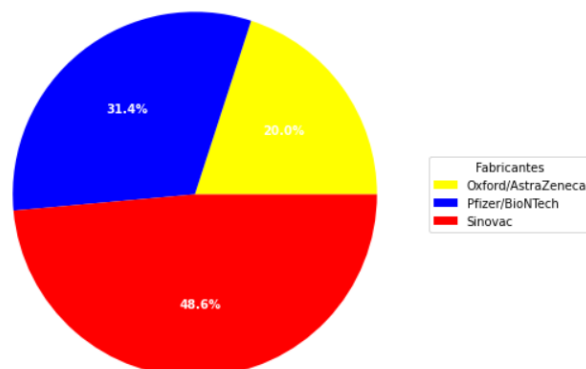
Ahora sumamos la cantidad de cada vacuna y lo filtramos por nombre y su cantidad total:

```
fabricantes=pd.read_csv('fabricantes.csv')
fabricantes=fabricantes.groupby(['vaccine']).sum()
fabricantes
```

Obtenemos como resultado los siguientes valores y gráfico (en la parte de la vacuna de Oxford es un 20%):

	total
vaccine	
Oxford/AstraZeneca	420000
Pfizer/BioNTech	659880
Sinovac	1020000

Fabricantes de vacunas





## Simulación

Tema: Simulación de Eventos.

Estudiante: Paulo Gonzalez

- **Generar histogramas de vacunas por mes de llega y fabricante.**

Para obtener la grafica de las vacunas se trabaja con los datos ya obtenidos anteriormente sobre la cantidad de las vacunas que se entregan mensualmente, hasta el mes actual (mayo) por lo cual se trabaja con la multiplicación de cada vacuna por 5.

```
fabricante = ['Oxford/AstraZeneca', 'Pfizer/BioNTech', 'Sinovac']
TotalPfizer = [0] * 5
TotalAstraZeneca = [0] * 5
TotalSinovac = [0] * 5

for i in range(len(fabricantesHist)):
    if(fabricantesHist.loc[i, "vaccine"] == fabricante[0]):
        TotalAstraZeneca[fabricantesHist.loc[i, "arrived_at"].month-1] += fabricantesHist.loc[i, "total"]
    elif(fabricantesHist.loc[i, "vaccine"] == fabricante[1]):
        TotalPfizer[fabricantesHist.loc[i, "arrived_at"].month-1] += fabricantesHist.loc[i, "total"]
    elif(fabricantesHist.loc[i, "vaccine"] == fabricante[2]):
        TotalSinovac[fabricantesHist.loc[i, "arrived_at"].month-1] += fabricantesHist.loc[i, "total"]

import matplotlib.pyplot as plt
import numpy as np

labels = ['Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo']

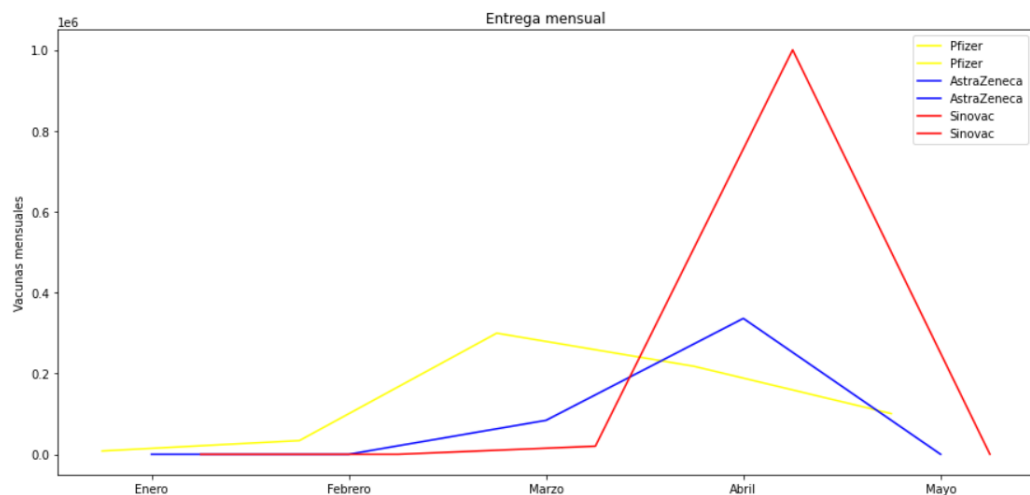
x = np.arange(5)

fig, ax = plt.subplots(figsize=(15,7))
V1 = ax.plot(x, TotalPfizer, 0.25, label='Pfizer', color="Yellow")
V2 = ax.plot(x + 0.25, TotalAstraZeneca, 0.25, label='AstraZeneca', color="Blue")
V3 = ax.plot(x + 0.5, TotalSinovac, 0.25, label='Sinovac', color="Red")

ax.set_ylabel('Vacunas mensuales')
ax.set_title('Entrega mensual')
ax.set_xticks(x+0.25)
ax.set_xticklabels(labels)
ax.legend()

plt.show()
```

Obteniendo como resultado:



- **Generar un reporte parametrizado que pueda ingresar los datos de las fechas inicio y fin para obtener la información de las gráficas vistas en el primer punto.** Para ello se ingresó los siguientes datos:

```
fechaInicio = "21/01/2021"
dias = 48
prediccionDia = "2021-05-30"
prediccionDiaFabricantes = "30/05/2021"
```



## Simulación

Tema: Simulación de Eventos.

Estudiante: Paulo Gonzalez

- **Generar un modelo matemático de predicción basado en regresión, de procesos de vacunación en base al número actual de vacunados (1 y 2 dosis) y a la llegada de nuevas vacunas.**

Para las predicciones se realizó el siguiente train y test para que se pueda realizar la predicción para las dos dosis, en cada una se adjunta el resultado obtenido.

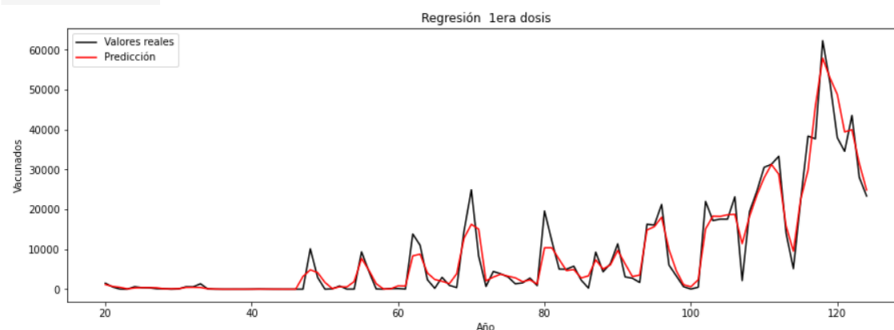
### 1. Dosis 1 (modelos de regresión)

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

dosisM1 = RandomForestRegressor(n_estimators=20, random_state=0)
dosisM1.fit(X_train, y_train.ravel())

y_pred1 = dosisM1.predict(X_test)
dosisPrediccion1 = pd.DataFrame({'Actual': y_test.flatten(), 'Prediccion': y_pred1.flatten()})
dosisPrediccion1.head()
```

	Actual	Prediccion
0	167933	158940.4
1	1962	4093.2
2	142782	153009.4
3	170276	168897.8
4	179292	181028.7



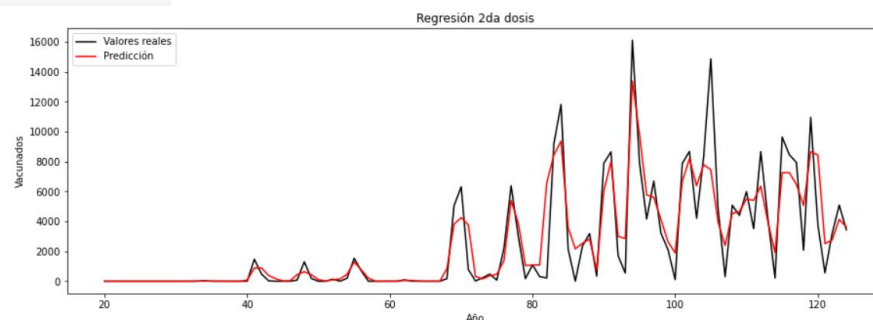
### 2. Dosis 2(modelos de regresión)

```
X_train, X_test, y_train, y_test = train_test_split(X, z, test_size=0.2, random_state=0)

dosisM2 = RandomForestRegressor(n_estimators=20, random_state=0)
dosisM2.fit(X_train, y_train.ravel())

y_pred2 = modeloRF2da.predict(X_test)
dosisPrediccion2 = pd.DataFrame({'Actual': y_test.flatten(), 'Prediccion': y_pred2.flatten()})
dosisPrediccion2.head()
```

	Actual	Prediccion
0	0	1.10
1	313	10380.75
2	0	470.85
3	222	7455.25
4	14873	18655.95





## Simulación

Tema: Simulación de Eventos.

Estudiante: Paulo Gonzalez

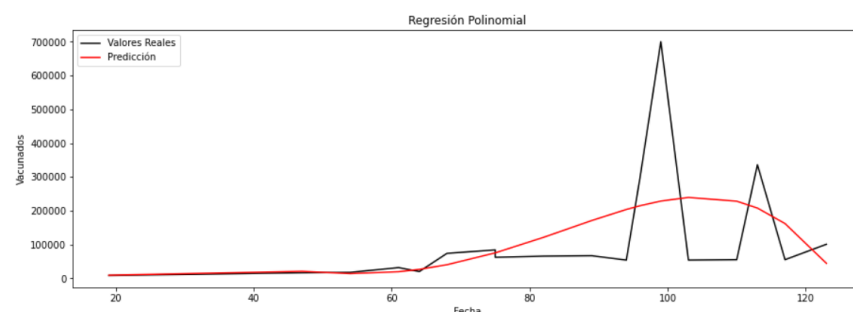
### 3. Regresión polinomial

Precisión: -9.47

```
fabricantes=fabricantes[['arrived_at','total']]
fabricantes

X = fabricantes.iloc[:, :-1].values
y = fabricantes.iloc[:, 1].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
poly_reg = PolynomialFeatures(degree=4)
X_poly = poly_reg.fit_transform(X)
pol_reg = LinearRegression()
pol_reg.fit(X_poly, y)
y_pred2 = pol_reg.predict(poly_reg.fit_transform(X_test))
preRF=r2_score(y_test,y_pred2)
print("Precisión del modelo: ", preRF)
```

Precisión del modelo: -9.47755846135113



### 4. Regresión Lineal

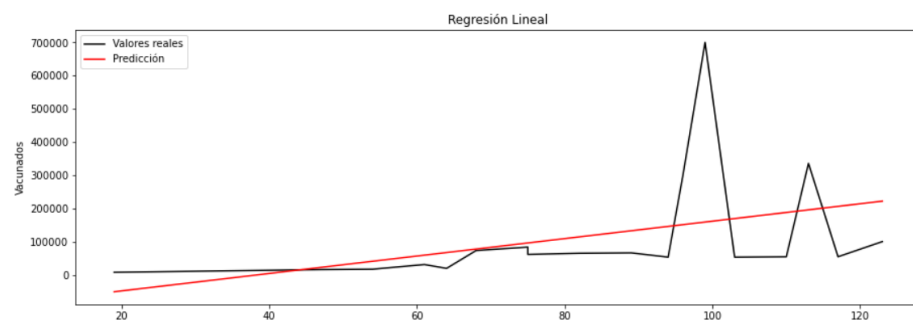
Precisión: -0.7675

```
x = fabricantes.iloc[:, :-1].values
y = fabricantes.iloc[:, 1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
b=regressor.intercept_
m=regressor.coef_
y_pred = regressor.predict(X_test)
preLin=r2_score(y_test,y_pred)
print("Precisión del modelo: ", preLin)
```

Precisión del modelo: -0.7675622720924429



Ahora comparando resultados tenemos que la regresión mas eficiente es la lineal por la cantidad de datos que se manejan a mas de progreso que tienen las llegadas de las nuevas vacunas, ahora bien, las predicciones para nuevas vacunas, para que todas las personas estén vacunadas con las primeras dosis y con las segunda hasta el 30/05/2021 son las siguientes:



## Simulación

Tema: Simulación de Eventos.

Estudiante: Paulo Gonzalez

```
date_string = prediccionDia
FMT = '%Y-%m-%d'
nuevaFecha = (datetime.strptime(date_string, FMT) - datetime.strptime("2021-01-01", FMT)).days
predVacunados = modeloRF1era.predict([[nuevaFecha]])

vacunados1era = pd.DataFrame({'Prediccion primera dosis': predVacunados.flatten()})
vacunados1era
```

Prediccion primera dosis

0	260175.6
---	----------

```
date_string = prediccionDia
FMT = '%Y-%m-%d'
nuevaFecha = (datetime.strptime(date_string, FMT) - datetime.strptime("2021-01-01", FMT)).days
predVacunados = modeloRF2da.predict([[nuevaFecha]])

vacunados2da = pd.DataFrame({'Prediccion segunda dosis': predVacunados.flatten()})
vacunados2da
```

Prediccion segunda dosis

0	3628.15
---	---------

```
date_string = "30/05/2021"
FMT = '%d/%m/%Y'
nuevaFecha = (datetime.strptime(date_string, FMT) - datetime.strptime("01/01/2021", FMT)).days
y_pred1 = regressor.predict([[nuevaFecha]])

predFab = pd.DataFrame({'Nuevas Vacunas hasta el 30/05/2021': [round(y_pred1[0])])
predFab
```

Nuevas Vacunas hasta el 30/05/2021

0	290595
---	--------

- Desarrollar y generar un proceso de comparación con al menos dos países (1. Latinoamérica, 1. Europa).

El país de Latinoamérica seleccionado es Chile y el país europeo es Inglaterra.

```
vacunacionChile=pd.read_csv('vacunacion_Chile.csv')
vacunacionEcuador=pd.read_csv('vacunas.csv')
vacunacionInglaterra=pd.read_csv('inglaterra2.csv',sep=';')
```

```
vacunacionInglaterra=vacunacionInglaterra.head(48)
vacunacionInglaterra
```

	areaType	areaName	areaCode	date	newVaccinesGivenByPublishDate	cumVaccinesGivenByPublishDate	newPeopleVaccinatedFirstDoseByPublishDate
0	overview	United Kingdom	K02000001	10/1/2021	NaN	2677971	NaN
1	overview	United Kingdom	K02000001	11/1/2021	165844.0	2843815	145076
2	overview	United Kingdom	K02000001	12/1/2021	223726.0	3067541	207661
3	overview	United Kingdom	K02000001	13/1/2021	288688.0	3356229	278945
4	overview	United Kingdom	K02000001	14/1/2021	321951.0	3678180	316694
5	overview	United Kingdom	K02000001	15/1/2021	345525.0	4023705	341317
6	overview	United Kingdom	K02000001	16/1/2021	298751.0	4322456	296131
7	overview	United Kingdom	K02000001	17/1/2021	192346.0	4514802	190107



## Simulación

Tema: Simulación de Eventos.

Estudiante: Paulo Gonzalez

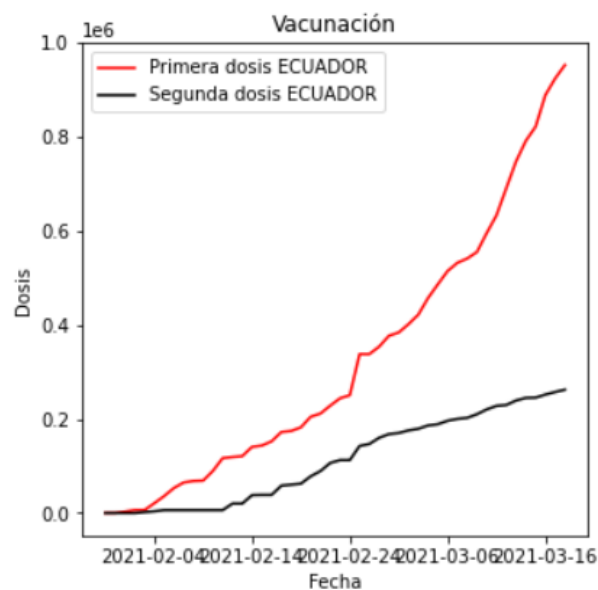
```
dias=48
inicio = datetime.strptime(x.iloc[vacunadosChile.loc[vacunadosChile['Fecha'] == '2021-01-30'].index[0]], '%Y-%m-%d')
fin = inicio + dt.timedelta(dias)
days = mdates.drange(inicio,fin,dt.timedelta(days=1))
plt.figure(figsize = (5,5))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=10))
plt.plot(days, y[:dias], label = 'Primera dosis CHILE', color='red')
plt.plot(days, z[:dias], label = 'Segunda dosis CHILE', color='black')
plt.title('Vacunación')
plt.xlabel('Fecha')
plt.ylabel('Dosis')
plt.legend()
plt.show()

dias=48
inicio = datetime.strptime(x.iloc[vacunadosChile.loc[vacunadosChile['Fecha'] == '2021-01-30'].index[0]], '%Y-%m-%d')
fin = inicio + dt.timedelta(dias)
days = mdates.drange(inicio,fin,dt.timedelta(days=1))
plt.figure(figsize = (5,5))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=10))
plt.plot(days, a[:dias], label = 'Primera dosis ECUADOR', color='red')
plt.plot(days, b[:dias], label = 'Segunda dosis ECUADOR', color='black')
plt.title('Vacunación')
plt.xlabel('Fecha')
plt.ylabel('Dosis')
plt.legend()
plt.show()

dias=48
inicio = datetime.strptime(x.iloc[vacunadosChile.loc[vacunadosChile['Fecha'] == '2021-01-30'].index[0]], '%Y-%m-%d')
fin = inicio + dt.timedelta(dias)
days = mdates.drange(inicio,fin,dt.timedelta(days=1))
plt.figure(figsize = (5,5))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=10))
plt.plot(days, c[:dias], '-', label = 'Primera dosis INGLATERRA', color='black')
plt.title('Vacunación')
plt.xlabel('Fecha')
plt.ylabel('Dosis')
plt.legend()
plt.show()

dias=48
inicio = datetime.strptime(x.iloc[vacunadosChile.loc[vacunadosChile['Fecha'] == '2021-01-30'].index[0]], '%Y-%m-%d')
fin = inicio + dt.timedelta(dias)
days = mdates.drange(inicio,fin,dt.timedelta(days=1))
plt.figure(figsize = (5,5))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=10))
plt.plot(days, d[:dias], '-', label = 'Segunda dosis INGLATERRA', color='red')
plt.title('Vacunación')
plt.xlabel('Fecha')
plt.ylabel('Dosis')
plt.legend()
plt.show()
```

- **Generar las graficas de regresión y comparar.**  
aquí podemos observar las gráficas comparativas

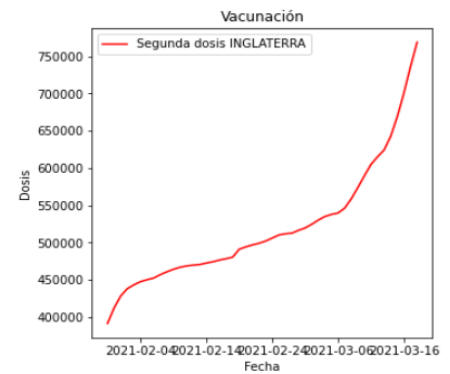
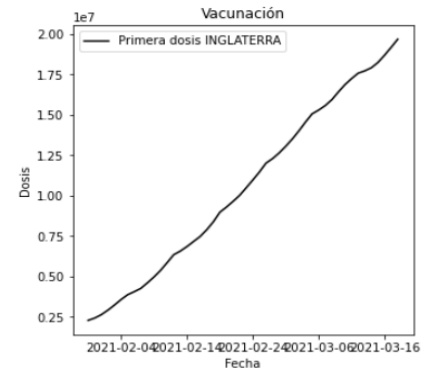
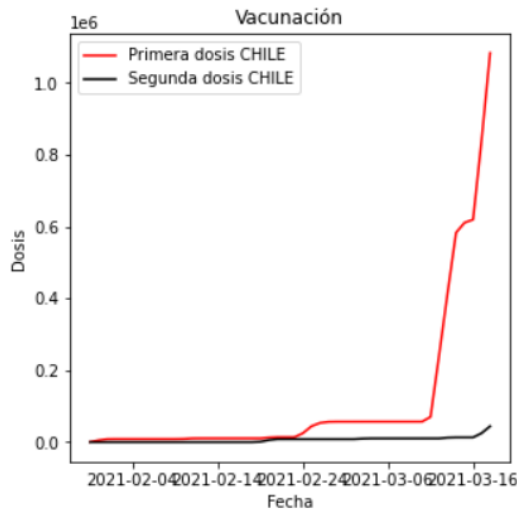




## Simulación

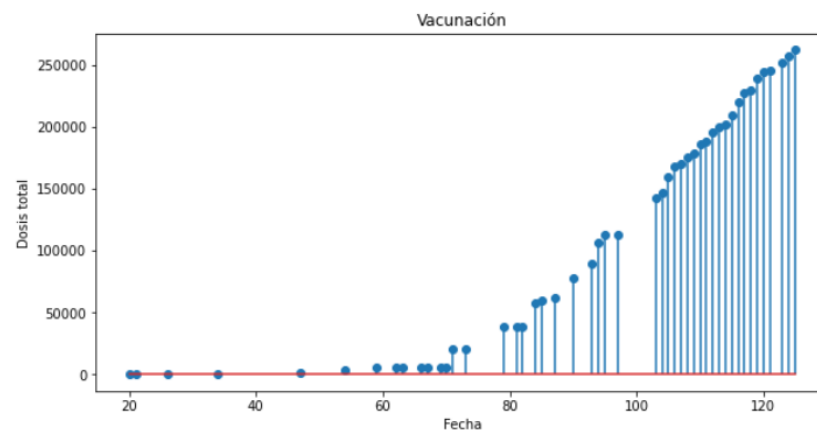
Tema: Simulación de Eventos.

Estudiante: Paulo Gonzalez



- Identificar cual es la fecha tentativa en la que todos los Ecuatorianos podrán ser vacunados con las dos dosis.

El proceso de vacunación ha ido avanzado lentamente, pero de una manera lineal podemos verificarlo con la siguiente grafica



Ahora generando un modelo de regresión polinomial podemos observar los siguientes datos y resultados en cuanto la fecha tentativa para la vacunación total para el país.





## Simulación

Tema: Simulación de Eventos.

Estudiante: Paulo Gonzalez

```
X = vacunas.iloc[:, :-1].values
y = vacunas.iloc[:, 1].values

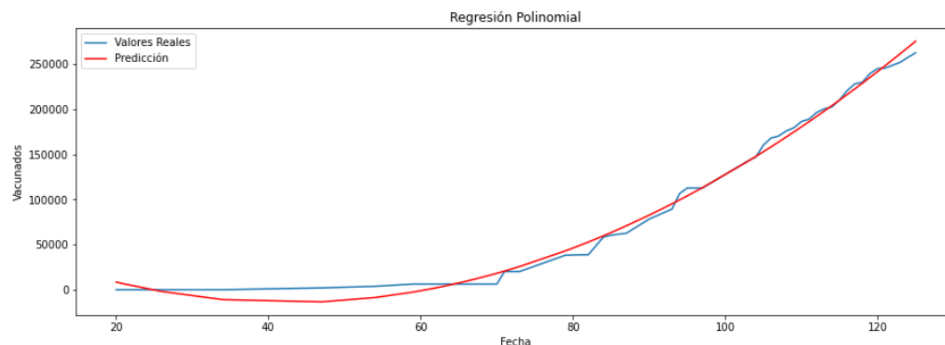
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=10)

poly_reg = PolynomialFeatures(degree=3)
X_poly = poly_reg.fit_transform(X)
pol_reg = LinearRegression()
pol_reg.fit(X_poly, y)

y_pred = pol_reg.predict(poly_reg.fit_transform(X_test))

predict = pd.DataFrame({'Actual': y_test, 'Prediccion': y_pred})
predict
```

	Actual	Prediccion
0	196350	192003.774098
1	106589	99469.189162
2	239431	235175.191209
3	227972	222433.252620
4	252044	261632.599943
5	62537	70751.217823
6	0	-10990.310564
7	170276	163619.316441
8	6228	1996.204188
9	6228	-2661.408048



Con esto podemos determinar según la cantidad de personas vacunadas y la población total del Ecuador, **la fecha tentativa es el 29/01/2023**

### ■ Conclusiones

Una ventaja muy notoria tiene el Random Forest a comparación de los otros modelos, además que nos permite trabajar con miles de variables es un modelo muy estable, pero cuando son pocos datos tiende a dar resultados muy poco favorables, eso se debe tomar muy en cuenta.

Gracias a lo realizado se puede obtener un panorama a futuro sobre nuestro país en cuanto a la pandemia, los gráficos y datos obtenidos nos permite dar un análisis de los posibles casos en los cuales se pueda mejorar todo.

### ■ Opinión

El proceso de vacunación dentro del país es demasiado tardío a comparación de otros países en donde evidentemente el tema del pandemia se tomo mucho mas en serio, y uno de los principales problemas fue la corrupción que existió dentro del mismo generando un caos y preocupación por como se lleva la emergencia sanitaria, pero viendo los resultados y predicciones propuesta podemos dar fe que la emergencia continuara, y se predijo en el caso que las vacunas sean importadas al país mensualmente y que sea así por casi 3 años, sería la única manera de que el país este casi inmune de esta enfermedad.



## **Simulación**

**Tema:** Simulación de Eventos.

**Estudiante:** Paulo Gonzalez



## Referencias:

[1]: <https://www.salud.gob.ec/plan-nacional-de-vacunacion-ecuador-2021-llega-segundo-lote-de-vacunas-pfizer-por-16-380-dosis/>

[2]: <https://www.edicionmedica.ec/secciones/salud-publica/-cuantas-personas-ya-se-han-vacunado-contr-la-covid-19-en-el-ecuador--97224>