

Creacion de un modelo de Regresión para predecir las personas que fallecen o se recuperan del COVID

Autor: Esteban David Rosero Perez

Generar un modelo de regresion de los casos confirmados de COVID-19 dentro del Ecuador el mismo que permita predecir el comportamiento y/o prediccion de la pandemia, tomar los datos desde el inicio de la pandemia e identificar etapas: Confinamiento, Toques de queda, Feriados, etc.

Librerias

```
In [192]: import pandas as pd
import numpy as np
from datetime import datetime, timedelta
from sklearn.model_selection import train_test_split
```

```
In [16]: from sklearn.linear_model import LinearRegression
from sklearn.pipeline import Pipeline
```

dataset

Dataset con los casos confirmados, muertos, recuperados por dia exacto.

```
In [20]: df = pd.read_csv('covid.csv')
df.sample(5)
```

```
Out[20]:
```

	date	confirmed	deaths	recovered	day
63	2020-05-03	29538	1564	3300	123
318	2021-01-13	224315	14229	193581	378
96	2020-06-05	41575	3534	20568	156
393	2021-03-29	325124	16746	281684	453
45	2020-04-15	7858	388	780	105

```
In [156]: df[df['day']==468]
```

```
Out[156]:
```

	date	confirmed	deaths	recovered	day
408	2021-04-13	347589	17351	298604	468

Modelo de Regresion

Primero procedemos a obtener los valores en los cuales van a ser sumamente importantes al momento de crear el modelo de regresion, en este caso nuestras variables seran los muertos que estan en la posicion [2:3] del dataset y como salida o featured obtenemos el día exacto que nos detallo la informacion de los muertos que esta en la posicion [4:5].

```
In [69]: featuresDead = df.iloc[:,4:5].values
targetDead = df.iloc[:,2:3].values
```

Para otro ejemplo vamos a predecir los recuperados vamos a obtener los valores en los cuales van a ser sumamente importantes al momento de crear el modelo de regresion, en este caso para los recuperados nuestras variables seran los que estan en la posicion [4:5] del dataset y como salida o featured obtenemos el día exacto que nos detallo la informacion de los recuperados que esta en la posicion [4:5].

```
In [73]: featuresRecovered = df.iloc[:,4:5].values
targetRecovered = df.iloc[:,3:4].values
```

Procedemos a realizar el modelo de regresion, primero debemos entrenarlo y para ello, debemos especificar cuales van a ser los valores de entrenamiento y de test, ingresando como variables el dataset requerido, en este caso tenemos el ejemplo del dataset de los muertos y de los recuperados.

```
In [82]: X_train, X_test, y_train, y_test = train_test_split(featuresDead, targetDead, test_size=0.2)
```

```
In [83]: model_dead = LinearRegression()
model_dead = model_dead.fit(X_train,y_train)
```

```
In [85]: X_train, X_test, y_train, y_test = train_test_split(featuresRecovered, targetRecovered, test_size=0.2)
```

```
In [86]: model_recovered = LinearRegression()
model_recovered = model_recovered.fit(X_train,y_train)
```

Mostramos el puntaje de validacion para el modelo de regresion en este caso para el modelo de los muertos es 0.96 y para los recuperados en 0.97.

```
In [84]: print("Dead linear Score : ",model_dead.score(X_test,y_test))
```

Dead linear Score : 0.9624268243887746

```
In [87]: print("Recovered linear Score : ",model_recovered.score(X_test,y_test))
```

Recovered linear Score : 0.9735775868999801

Nuevas Pruebas

Para realizar las pruebas procedemos a utilizar los dos datasets que tenemos, el de recuperados y el de los muertos en este caso vamos a predecir los siguientes días del dataset, desde el día 14-04-2021.

```
In [182]: data=[]
dead=[]
recovered=[]
for x in range(469,479):
    data.append(x)
    dead.append(round(model_dead.predict(np.array(x).reshape(1,-1))[0][0]))
    recovered.append(round(model_recovered.predict(np.array(x).reshape(1,-1))[0][0]))
```

```
In [183]: inicio = datetime(2021,4,14)
fin = datetime(2021,4,23)
lista_fechas = [(inicio + timedelta(days=d)).strftime("%Y-%m-%d")
for d in range((fin - inicio).days + 1)]
```

Aqui se puede observar el modelo de prediccion para los muertos y los recuperados, en las siguientes fechas posteriores al dataset

```
In [184]: dfnew = pd.DataFrame(list(zip(dead,recovered)), columns = ['dead','recovered'])
dfnew['date']=lista_fechas
dfnew
```

```
Out[184]:
```

	dead	recovered	date
0	18819.0	270847.0	2021-04-14
1	18868.0	271618.0	2021-04-15
2	18916.0	272388.0	2021-04-16
3	18964.0	273159.0	2021-04-17
4	19012.0	273930.0	2021-04-18
5	19061.0	274701.0	2021-04-19
6	19109.0	275472.0	2021-04-20
7	19157.0	276243.0	2021-04-21
8	19205.0	277014.0	2021-04-22
9	19253.0	277784.0	2021-04-23

Porcentaje de Crecimiento por dia

Aqui podemos observar que en el dataset en los muertos tiene un crecimiento diario de 0.25 respectivamente, y en los recuperados hay un crecimiento diario de 0.28 respectivamente

```
In [188]: porcDead=[]
for d in range(0,9):
    porcDead.append((100*(dead[d]-dead[d+1]))/dead[d+1])
porcDead
```

```
Out[188]: [-0.2596989612041552,
-0.2537534362444491,
-0.2531111579835478,
-0.25247212286976645,
-0.2570694087403599,
-0.2511905384897169,
-0.25056115258130185,
-0.2499349127831294,
-0.24931179556432764]
```

```
In [189]: porcRec=[]
for d in range(0,9):
    porcRec.append((100*(recovered[d]-recovered[d+1]))/recovered[d+1])
porcRec
```

```
Out[189]: [-0.28385453099573665,
-0.28268499346520404,
-0.28225319319517206,
-0.28145876683824333,
-0.28066879989515875,
-0.27988325492246036,
-0.27910209489471227,
-0.2783252831986831,
-0.27719379085908474]
```

Conclusiones

El modelo de regresion nos ayuda mediante dos variables, buscar una relacion entre las variables y hace una serie de pruebas para validar esas relaciones. con la ayuda de este modelo de regresion con las dos variables que le probamos, por ejemplo la informacion de los muertos fue una variable de salida y la relacion de la fecha era otra variable pero de entrada en la cual el modelo nos ayuda en la prediccion de nuevas fechas de entrada. Existen modelos en el cual el promedio de efectividad es mejor al modelo de regresion lineal que utilizamos en este ejemplo.

Observaciones

El modelo de regresion se puede mejorar, pero existen modelos que se adaptan mejor segun el numero de datos de entrada y salida, las variables son muy importantes para el entrenamiento de los modelos por lo cual se debe tener en cuenta que tipo de variables se van a manejar.

In []: