Escuela Politécnica Nacional del Ecuador
Danny Díaz Probabilidad y Estadística
Segundo Semestre

Importancia de la Estadística en diferentes ámbitos de la vida cotidiana.

Inteligencia artificial y redes neuronales artificiales

La recomendación de datos y productos que hacen las grandes empresas están basadas en frecuencias de consumo y sobre todo probabilidades que estima respecto de la media. Una red neuronal artificial es la encargada de recolectar el historial de las personas y 'entrenarse' para predecir qué es lo más repetitivo (cual categoría) y así nos puede dar predicciones de lo siguiente que querrán buscar las personas.

Las inversiones en bolsa

La importancia de la estadística y su aplicación a la hora de diseñar sistemas de trading está sumamen (Santillan, 2013)te demostrada. De hecho, la estadística es junto a la información del volumen de negociación, dos pilares fundamentales a la hora de adaptarse a los cambios que se suelen presentar en los mercados.
Cuando vemos activos que se encuentran subiendo de manera clara y vertical una de las preguntas que más nos hacemos es ¿Hasta dónde va a subir este activo? **¿Cuándo tendrá un descanso para incorporarnos a su escalada de precios?**

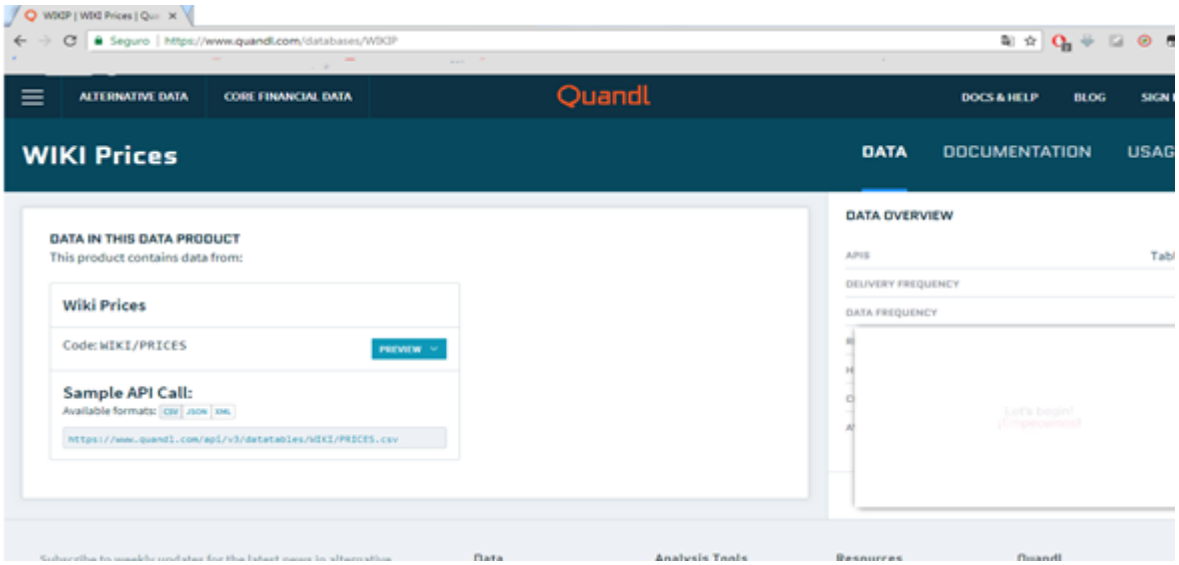
Los **activos cotizados se mueven con formaciones de dientes de sierra**, es decir, si nos encontramos en una tendencia alcista, **a cada subida le acompaña una corrección** y si nos encontramos en una tendencia bajista, **a cada tramo impulsivo bajista le sigue una corrección al alza** (García, 2013)**.**

En el lenguaje de programación Python podemos hacer un modelo de regresión logística (y/o lineal) en el que podemos predecir los precios de cierta organización (Google) programa que implementé y se encuentra en la siguiente página de este documento.

Escuela Politécnica Nacional del Ecuador
Danny Díaz Probabilidad y Estadística
Segundo Semestre

Distribución normal, regresión lineal, probabilidades e Inteligencia artificial

Como estudiante de la facultad de ingeniería en Sistemas, he implementado un programa en el lenguaje de programación 'python' el cual accede a una base de datos pública (Quandl) a los precios de ajuste que tienen las acciones de Google.



Usando:

```
: quandl.get('WIKI/GOOGL')
```

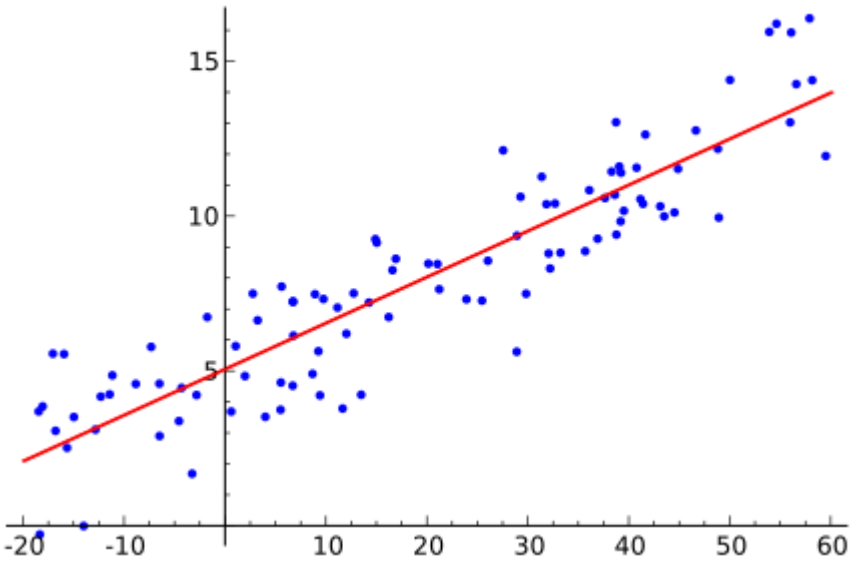
Obtenemos los datos que requerimos.

```
"C:\Program Files (x86)\Python36-32\python.exe" "C:/Users/User/Documents/le
               Open     High      Low    Close     Volume  Ex-Dividend  \
Date
2018-02-06  1033.98  1087.38  1030.01  1084.43  3732527.0          0.0
2018-02-07  1084.97  1086.53  1054.62  1055.41  2544683.0          0.0
2018-02-08  1059.87  1063.93  1005.12  1005.60  3067173.0          0.0
2018-02-09  1025.88  1051.72   997.00  1043.43  4436032.0          0.0
2018-02-12  1056.67  1065.57  1045.49  1054.56  2796258.0          0.0

            Split Ratio  Adj. Open  Adj. High  Adj. Low  Adj. Close  \
Date
2018-02-06          1.0    1033.98    1087.38   1030.01     1084.43
2018-02-07          1.0    1084.97    1086.53   1054.62     1055.41
2018-02-08          1.0    1059.87    1063.93   1005.12     1005.60
2018-02-09          1.0    1025.88    1051.72    997.00     1043.43
2018-02-12          1.0    1056.67    1065.57   1045.49     1054.56

            Adj. Volume
Date
2018-02-06    3732527.0
2018-02-07    2544683.0
2018-02-08    3067173.0
2018-02-09    4436032.0
2018-02-12    2796258.0
```

Podemos ver los precios de ajuste de los últimos cinco días. Pero la base de datos tiene un sin número de información que data desde el 2004.

El algoritmo lo que hace es tomar un 80% de todos los datos de la base de datos desde aproximadamente el año 2004, hasta el día de hoy, luego intenta crear una línea recta, que le permita obtener los valores reales (importante) de ajuste de ese 80% de los datos. Es decir, realiza una regresión lineal. Es una manera de entrenarse para entender lo que esta pasando en el mundo que le rodea, sus redes neuronales adquieren información comparando entre datos reales y datos realizados como predicción.

¿Qué es regresión Lineal?



El análisis de regresión lineal es una técnica estadística utilizada para estudiar la relación entre variables. Se adapta a una amplia variedad de situaciones. En la investigación social, el análisis de regresión se utiliza para predecir un amplio rango de fenómenos, desde medidas económicas hasta diferentes aspectos del comportamiento humano. En el contexto de la investigación de mercados puede utilizarse para determinar en cuál de diferentes medios de comunicación puede resultar más eficaz invertir; o para predecir el número de ventas de un determinado producto (Marin, 2008).

Aunque un diagrama de dispersión permite formarse una primera impresión muy rápida sobre el tipo de relación existente entre dos variables, utilizarlo como una forma de *cuantificar* esa relación tiene un serio inconveniente: la relación entre dos variables no siempre es perfecta o nula; de hecho, habitualmente no es ni lo uno ni lo otro.

La regresión lineal nos permite encontrar una recta que mejor se ajuste a esa dispersión de los datos.

Cuando graficábamos una recta en el plano cartesiano, en una relación dependiente de ‘Y’ y ‘X’. Podíamos encontrar cualquier punto f(x) que quisiéramos siempre y cuando tuviéramos la ecuación de la recta y=mx+b, pues bien, en regresión lineal ocurre algo parecido, es solo que los parámetros pasan a ser más complejos que una simple pendiente y punto de corte y de hecho en la regresión más simple suele hacerse un ajuste promedio

de sumatorios de cada posible pendiente, puntos de corte y todo lo que es importante al analizar el conjunto de datos.

Volviendo al análisis del programa, realiza una regresión lineal:

```
tipo_clasificador = LinearRegression()

tipo_clasificador.fit(X_train, y_train)
```

Y usa los datos de los que ya entreno nuestro algoritmo.

Y así es capaz de regresarnos una predicción de los futuros 34 días, en base al 80% de los datos que analizo del pasado:

```
predicciones de los precios de cierre para los siguientes 34 dias:
[ 1092.15032255  1089.1443223   1083.03184979  1078.64621428  1076.2364631
  1095.30081307  1113.83473701  1118.75522018  1133.54264285  1137.63933473
  1136.47806712  1133.78016926  1135.65388709  1153.67325302  1153.46537527
  1163.29543166  1159.98334664  1167.60612106  1187.65049874  1200.40733298
  1195.05436333  1206.73727295  1212.36603549  1210.42746062  1201.13617178
  1206.75535289  1205.26874256  1141.00388023  1088.11896096  1104.4221316
  1076.38912423  1023.84363212  1063.68470651  1076.21911972]
La presicion del algoritmo es de:0.975634054438
```

La precisión del algoritmo

Había dicho que solo tomamos el 80% de los datos, ya que el restante será usado para ‘testear’ que tan buena es la línea recta que creamos, es decir, usamos datos reales que no le mostramos nunca al programa e intentamos que ‘adivine’ a ver si coinciden los datos reales con las predicciones.
La precisión del algoritmo nos dice que tanto acertó y que tanto se equivocó con la predicción de ese 20% restante de datos de las fechas más recientes.
Para que sea más visual, invocamos los últimos cinco días, las últimas cinco predicciones:

```
Adj. Close  HLC_change  CLO_change  Adj. Volume  \
       NaN         NaN         NaN          NaN
       NaN         NaN         NaN          NaN
       NaN         NaN         NaN          NaN
       NaN         NaN         NaN          NaN
       NaN         NaN         NaN          NaN

Futuro Adj. Close   Prediccion
              NaN  1104.422132
              NaN  1076.389124
              NaN  1023.843632
              NaN  1063.684707
              NaN  1076.219120
```

Son las predicciones de precio para los días 30, 31, 32, 33, 34 después del día de hoy.

Distribución normal ESTÁNDAR en el algoritmo

Algo importante a notar es:

```
X = preprocessing.scale(X)
```

En el código, esto significa que los datos que tenemos son normalizados, es decir, debido a las gigantes diferencias que pueden haber entre los datos de la tabla, es necesario escalarlos, y la mejor manera es ubicando la media de todos en cero y su varianza en uno. Es aplicada una distribución normal estándar de los datos para poder trabajar más eficientemente con ellos.
La gráfica de las predicciones nos queda así:

El color azul es lo que el código predijo.
Podemos notar una tendencia alcista con algunas correcciones de precios.

Programa fuente

Código fuente (Python y librerías especiales: numpy, sklearn, matplotlib,quandl):

```
import quandl
import math
import numpy as np
from sklearn import preprocessing, model_selection
from sklearn.linear_model import LinearRegression
import datetime
import matplotlib.pyplot as plt
from matplotlib import style
style.use('ggplot')
# Base de datos quandl, obtenemos algunos precios para practicar
df = quandl.get('WIKI/GOOGL')
print(df.tail())
# seleccionamos los precios de ajuste que creemos convenientes
df = df[['Adj. Open', 'Adj. High', 'Adj. Low','Adj. Close', 'Adj. Volume', ]]
# Resultado: Adj. High es 'x' porcentaje (de Adj. Close) mas que Adj. Close.
df['HrC_change'] = (df['Adj. High'] - df['Adj. Close']) / df['Adj. Close'] * 100.0
# Resultado: Adj. Close es 'x' porcentaje (de Adj. Open) mas que Adj. Open
df['CrO_change'] = (df['Adj. Close'] - df['Adj. Open']) / df['Adj. Open'] * 100.0
df = df[['Adj. Close', 'HrC_change', 'CrO_change', 'Adj. Volume', ]]
# Esta es la columna que hara la prediccion
columna_prediccion = 'Adj. Close'
# Realizamos una limpieza a la base de datos, primero llenamos los valores no asignados
# Aqui podriamos tambien ( y es recomendado) usar la media de la columna para rellenar
los datos faltantes
df.fillna(-99999, inplace=True)
# definimos a la prediccion como el 1% de todos los datos
prediccion = int(math.ceil(0.01 * len(df)))
etiqueta_del_futuro = 'Futuro ' + columna_prediccion
df[etiqueta_del_futuro] = df[columna_prediccion].shift(-prediccion)
# tomaremos todas las caracteristicas excepto la columna creada reientemente
# X: es todos el frame excepto la columna 'etiqueta_del_futuro'

X = np.array(df.drop([etiqueta_del_futuro], 'columns'))  # 'columns' = 1
X = preprocessing.scale(X)  # escala los datos, haciendo la media cero y la varianza 1.

DISTRIBUCION NORMAL
X_lately = X[-prediccion:]  # no consideraremos los valores no asignados
X = X[:-prediccion]
# y: toma los futuros precios
df.dropna(inplace=True)
y = np.array(df[etiqueta_del_futuro])
# y entrena X, y testea a X
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y, test_size=0.2)
# llenamos el modelo lineal
tipo_clasificador = LinearRegression()
tipo_clasificador.fit(X_train, y_train)
presicion = tipo_clasificador.score(X_test, y_test)



conjunto_predicciones = tipo_clasificador.predict(X_lately)
print("predicciones de los precios de cierre para los siguientes 34 dias: ")
print(conjunto_predicciones,"\nLa presicion del algoritmo es de:" + str(presicion))
df['Prediccion'] = np.nan # predicciones
anterior_fecha = df.iloc[-1].name
```

```
anterior_unix = anterior_fecha.timestamp()
un_dia_en_segundos = 86400
siguiente_unix = anterior_unix + un_dia_en_segundos

for i in conjunto_predicciones:
        siguiente_fecha = datetime.datetime.fromtimestamp(siguiente_unix)
        siguiente_unix += un_dia_en_segundos
        df.loc[siguiente_fecha] = [np.nan for _ in range(len(df.columns) - 1)] + [i]
print(df.tail())
df['Adj. Close'].plot()
df['Prediccion'].plot()
plt.legend(loc=4)
plt.xlabel('Datos')
plt.ylabel('Precio')
plt.show()
```

```
info_columns = {

        'HrC_change': 'High respect Close, cambio de porcentaje',

        'CrO_change': 'Close respect Open, cambio de porcentaje',

        'etiqueta_futuro': 'datos del futuro',

}
```

Conclusiones:

- Ø La probabilidad nos permite modelar eventos y fenómenos de futuro.
- Ø La distribución normal, nos permite escalar los datos para que tengan un media de 0 y varianza de 1. Importante cuando se trabaja con datos con diferencias enormes.
- Ø La regresión lineal permite obtener una relación entre todos los datos dispersos (incluyendo los atípicos).

Ø La inteligencia artificial se basa en gran parte de la probabilidad y estadística, que determinará (mediante entrenamientos con la estadística como vimos en este ejemplo) cual es el mejor camino o decisión a tomar.

Recomendaciones:

-> Si se quiere replicar el código tomar en cuenta que la base de datos posee algunos días sin ninguna información, se debe realizar un 'drop' y considerar cada día eliminado para que así el modelo de regresión lineal no tenga tantos vacíos.
-> Al trabajar con una base de datos gigante, siempre habrá algunos datos vacíos ya sea por error del recolector de información o pérdida digital, para esto, debemos recurrir a la media y rellenar los datos faltantes con ella, así evitamos perder posible valiosa información que se encuentra en otras columnas.

Bibliografía:

García, J. L. (17 de 09 de 2013). *Finanzas.* Obtenido de finanzas web site: http://www.finanzas.com/noticias/mercados/bolsas/20130917/como-utilizar-proyecciones-fibonacci-2477509.html

Marin, J. M. (2008). *Universidad Carlos III de Madrid.* Obtenido de UC3M web site: http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/GuiaSPSS/18reglin.pdf

Santillan, M. L. (3 de JULIO de 2013). *Unam.* Obtenido de Fundacionunam web site: http://www.fundacionunam.org.mx/humanidades/la-estadistica-en-nuestra-vida-diaria/

(Adicional) Curso de sklearn y machine learning con Python: https://www.youtube.com/playlist?list=PLQVvvaa0QuDfKTOs3Keq_kaG2P55YRn5v

Base de datos Quandl: https://www.quandl.com/databases/WIKIP