6PDE Machine Learning con Python **(Fecha: Febrero 2020)**

Evaluación del modulo II

Profesor: Abraham Zamudio Ch. *Modulo:* II :sympy,numpy,scipy,matplotlib,pandas

PREPROCESAMIENTO DE DATOS CON PYTHON

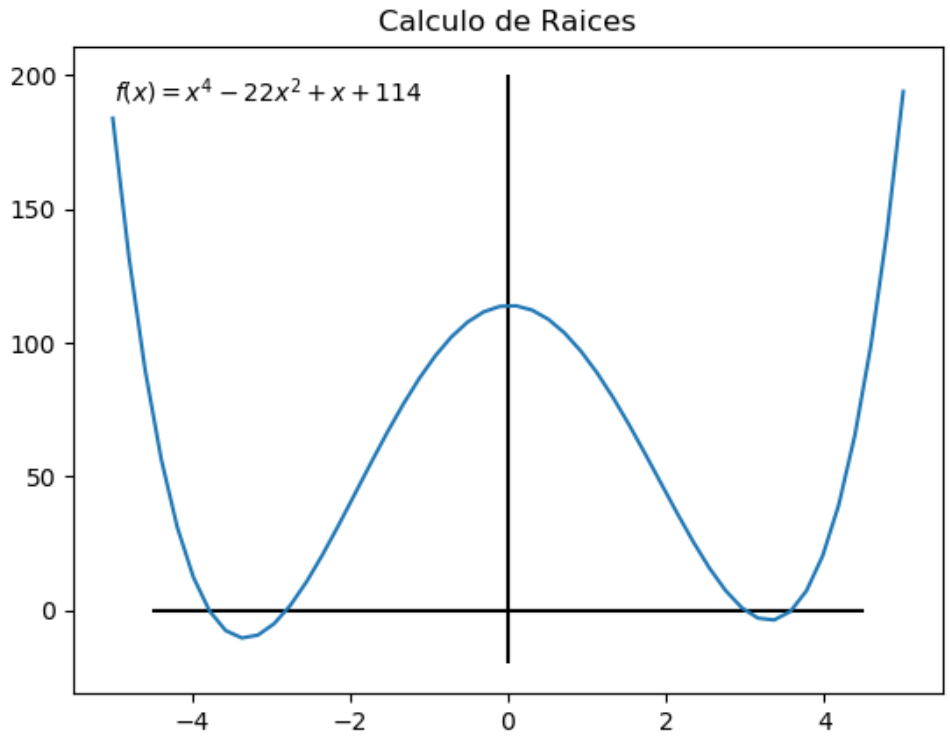
1. El *Ratio de sharpe* es una medida de compensación entre el beneficio (exceso de rendimiento) y el costo (riesgo total) para una inversión como una cartera. Escriba un programa de Python para estimar el ratio de sharpe aplicando la siguiente fórmula:

$$\text{Sharpe} = \frac{\bar{R} - \bar{R}_f}{\sigma} \tag{0.1}$$

   En la expresión anterior $\bar{R}$ es el rendimiento medio de la cartera, $\bar{R}_f$ es la tasa media libre de riesgo y σ es el riesgo de la cartera.En el script (o programa) a desarrollar se debe pedir al usuario el ingreso de estas tres variables por teclado.

2. Usando el modulo `sympy` defina la siguiente función simbólica:

$$f(x) = 114 - 22x^2 + x^4 + x \tag{0.2}$$



- Encuentre las raíces del polinomio anterior.Tenga en cuenta la función `solve`[1] del modulo `sympy`.
- Use `scipy` para calcular las raíces[2] del polinomio.
- Use `scipy` para calcular los mínimos[3] del polinomio.

3. La siguiente función sirve para generar una lista de números pseudoaletorios.

```
def gen_rnd(n):
        """
        Parametros
        ----------
                n : Variable de tipo Entero
                La funcion gen_rnd recibe como unico
                argumento un numero entero
        Salida
        -------
                List1 : Variable de tipo Lista
                La salida de la funcion es una lista de
                numeros enteros pseudoaleatorios.

        """
        List1 = []
        yi = int(np.random.randint(1,124578,1))
        for i in range(n):
                yi = (124578*yi+1)%(125)
                List1.append(yi)
        return List1
```

Genere una lista de 500 elementos usando la función `gen_rnd`.Notar que muy probablemente esta lista contenga muchos elementos repetidos.

```
Lista1 = gen_rnd(500)
```

Se pide resolver los siguientes items :

a) Escriba una función que devuelva una lista con los elementos únicos de la variable `Lista1`. **NO** esta permitido usar la función `unique` del modulo `numpy`, sin embargo puede servir para corroborar su respuesta

b) Cree una lista con todos los elementos de `Lista1` que sean múltiplos de 7 o múltiplos de 13.

c) Escriba una función que verifique que cada uno de los elementos de `Lista1` sea un numero primo o no.Esta función debe devolver un diccionario en el que cada llave sea un elemento de `Lista1` y su correspondiente valor sea TRUE (en caso el numero sea primo) y FALSE (en caso el numero no sea primo).

[1] https://docs.sympy.org/latest/modules/solvers/solvers.html
[2] https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.optimize.fsolve.html
[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html

4. Escribe una función que genere contraseñas, el único argumento de la función debe ser la longitud (cantidad de caracteres mayor a 9) de la contraseña.

   Sea creativo con la forma en que genera las contraseñas: las contraseñas seguras tienen una combinación de letras minúsculas, letras mayúsculas, números y símbolos. Las contraseñas deben ser aleatorias, generando una nueva contraseña cada vez que el usuario solicite una nueva contraseña. Los posibles elementos a usar en la contraseña provienen de lista `Caracteres`

   ```
   1 Caracteres = [chr(i) for i in range(128)][33:-1]
   ```

   ```
   ['!', '"', '#', '$', '%', '&', "'", '(', ')', '*', '+', ',', '-', '.', '/', '0',
    '1', '2', '3', '4', '5', '6', '7', '8', '9', ':', ';', '<', '=', '>', '?', '@',
    'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',
    'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '[', '\\',']', '^', '_', '`',
    'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
    'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', '{', '|', '}', '~']
   ```

5. Considere los archivos de datos `nombres_CA.txt` y `nombres_TX.txt`, en el cual se listan los nombres (variable `Name`) de los niños/niñas nacidos en las ciudades *CA* y *TX* entre los años (variable `year`) de 1910 y 2018. Ademas de la cantidad de veces que estos nombres se repiten (variable `Num`).Escriba un script en `python` para dar respuesta a las siguientes cuestiones :

   a) Escriba una función que calcule cuantos bebes nacieron por año para cada una de las ciudades.

   b) Escriba una función para calcular cuantos bebes nacieron de sexo femenino (variable `Sex` = F) entre un rango de años. La función debe ejecutarse así : `f1(year_min, year_max)`

   c) Escriba una función que retorne los 3 nombres mas comunes en cada una de las ciudades.

   d) Escriba una función que calcule el segundo nombre mas popular en cada una de las décadas en que se tiene la data (1910-2018).

6. Considere el dataset `cherry.csv` cargandolo en memoria con la siguiente instruccion

   ```
   import pandas as pd
   cherry = pd.read_csv("cherry.csv", sep=";")
   ```

   Notar que las variables son `Girth`,`Height` y `Volume`. Resuelva las siguientes cuestiones usando código `python` :

   a) ¿Cual es la cantidad de filas del dataset?

   b) Haga un diagrama de dispersión para cada una de las 3 variables.

   c) Realiza un boxplot para cada una de las variables. Cuales son los valores de las variables `Girth` y `Height` para el outlier de la variable `Volume`.