Software matemático especializado en el cálculo de la correlación de la distancia para bases de datos y funciones

Andrés Rubiano, Jorge Camacho, Santiago Mariño.

No. de Equipo Trabajo: 9

INTRODUCCIÓN

En este proyecto se busca optimizar el cálculo de la *correlación de la distancia* a través del lenguaje de programación *Python* usando como herramienta las diferentes estructuras de datos con las que contamos hoy en día. En este programa dichas herramientas serán utilizadas para crear una interfaz gráfica con la cual el usuario tendrá una mayor facilidad al representar las muestras que el usuario necesita ingresar para calcular la *correlación de la distancia*.

Calcular este valor es algo sencillo temiendo en cuenta las herramientas computacionales con las que contamos hoy en día, pero debido a la gran cantidad de datos que se pueden llegar a tener, se presentaran complicaciones a la hora de ejecutar el programa, más específicamente, en el tiempo de ejecución de este, de ahí la importancia de optimizar este proceso.

PLANTEAMIENTO DEL PROBLEMA

El cálculo de la *correlación de la distancia* se puede realizar a través del paquete *energy* escrito en el lenguaje R. Lo que se busca en este proyecto es implementar el cálculo de este en el lenguaje de programación *Python*, cambiando la forma en las cuales se almacenan los datos experimentales que se tienen, esto se realizara con la ayuda de las estructuras de datos que conocemos(*Listas, Colas, Pilas, Arboles, Hash*) y así, utilizando distintos datos experimentales, se obtendrá el tiempo que se tarda en calcular la *correlación de la distancia* y se podrá escoger la forma más óptima de guardar los datos para realizar el cálculo.

USUARIOS DEL PRODUCTO DE SOFTWARE

El público al que está dirigido el software es en su mayoría estadísticos, matemáticos, ingenieros y economistas, entre otros, puesto que estos en el desarrollo de sus carreras o en el campo laboral utilizan aplicaciones y software de análisis, bases y estructuras de datos, en las cuales pueden llegar a necesitar qué relación hay entre los datos con los cuales están trabajando, he aquí la importancia de la *correlación de la distancia*.

REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

- Los datos que se ingresan al software, son tomados de bases de datos, En nuestro caso poseemos datos aleatorios que se usaron para las pruebas de tiempo y se encuentran formato .csv, además tenemos datos relacionados con el CoVid-19 también en este formato, por lo tanto, es necesario que se puedan leer archivos en formato .csv.
- La aplicación almacena los datos suministrados en matrices, las cuales son implementadas con arreglos y listas enlazadas, a estas implementaciones se les compara el rendimiento más adelante para así optimizar el proceso del cálculo.
- Los reportes que arroja la aplicación son el coeficiente correlación de distancia y la gráfica interactiva de datos.

INTERACCIONES CON EL USUARIO

- **Ingreso de base de datos por el usuario**: El usuario puede ingresar su base de datos desde el directorio, en donde esté se encuentre guardado, este debe estar en formato .csv.

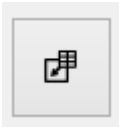


Figura 1

- **Calcular la *correlación de la distancia* con muestras aleatorias:** con este botón (Figura 2) se realiza el cálculo de la *correlación de la distancia* usando datos aleatorios, este se almacena en arreglos o listas enlazadas del tamaño que el usuario desee en el apartado “Tamaño de la muestra”, si el usuario no ha ingresado ningún valor antes de presionar el botón superior se generara una ventana de error.

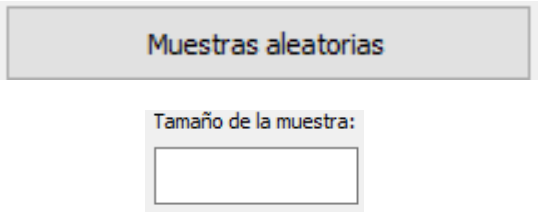


Figura 2

- **Calcular la *correlación de la distancia* con muestras de archivos .csv:** con el botón (Figura 3), este proceso se hará con el archivo que se importó con la Figura 1, si este no ha sido importado se genera una ventana de error.

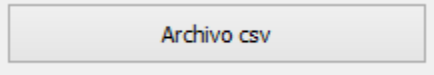


Figura 3

- **Graficar los datos:** También se podrá visualizar el comportamiento de los datos. Con en el botón (Figura 4), el usuario podrá graficar su conjunto de puntos para el respectivo análisis, esta generara una ventana de error si no se ha calculado la correlación de la distancia de manera previa con alguno de los dos métodos anteriores.

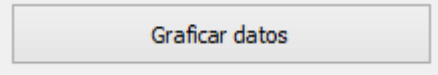


Figura 4

- **Modo de almacenamiento:** Con los modos de ingreso (Figura 5), el usuario podrá seleccionar si almacena los datos por listas o por arreglos.

Figura 5

- **Cálculo y gráfico de una función matemática:** Por medio de un campo de texto denominado "Ingresar función por pantalla (Figura 6) se puede introducir la función matemática (Polinomios, Racionales, Exponenciales), con el apartado de "Dominio" se introducirán los valores del "Límite inferior", "Límite superior" y "Salto" (espacio entre dato y dato en el intervalo) respectivamente para determinar el intervalo en que se va a trabajar la función, en el momento en el que se desee graficar la función dada y calcular su correlación de la distancia se confirmará con el botón ilustrado a la derecha. Todo lo anterior se puede realizar gracias a que los datos se almacenan en arreglos a través de un árbol binario, el uso del primer campo puede generar errores, para esto se deben agregar paréntesis de una forma estratégica hasta que el programa arroje la función deseada, además si falta algún campo por ser llenado y se presiona el botón de la derecha se generara una ventana de error.

Figura 6

- **Undo y Redo:** Para retroceder o avanzar sobre las acciones realizadas (Figura 7), esto se realizó con la implementación de pilas, el abuso de estas puede generar errores, por lo tanto, es mejor volver a realizar el proceso si se presentan problemas.

Figura 7

- **Prioridad de funciones matemáticas**: Para agregar acciones a la cola de prioridad es necesario rellenar los campos de la Figura 6, primero se asigna una prioridad en el apartado "prioridad", cuando este esté lleno (si no está lleno este campo y se desean agregar datos a la cola se genera una ventana de error) se agrega a la cola de prioridad con el botón que se encuentra al lado, finalmente para ejecutar la función con mayor prioridad se desencola con el botón que se encuentra a la derecha, esto fue implementado con una cola de prioridad.

Figura 8

- **Covid-19:** Con este botón (Figura 9) se implementa un hash el cual toma de una base datos las fechas en las cuales se tomaron datos relacionados con el Covid-19, en nuestro caso con la clave que es la fecha se guardan en dos arreglos el número de casos totales y el número de muertos de ese día cada uno en un arreglo diferente, después de esto se genera la *correlación de la distancia* de los datos y se grafican para ver si estos están relacionados o no lo están.

Figura 9

REQUISITOS FUNCIONALES MINIMOS DEL SISTEMA

- Mostrar la ubicación del archivo donde se encuentra la base de datos.
- Graficar los datos en diagramas de dispersión.
- Mostrar el cálculo de la correlación de la distancia
- Permitir almacenar los datos en listas o en arreglos.
- Permitir el ingreso de datos en formato CSV a través de la librería pandas.
- Representación gráfica de una función matemática.
- Cálculo de la correlación de la distancia para una función matemática.
- Retroceder acciones (Undo).
- Rehacer acciones (Redo).
- Añadir acciones con prioridades dadas a una cola.
- Tomar datos de Covid-19 (Casos por día vs muertes) y ver su relación gráficamente

INTERFAZ GRAFICA DEL USUARIO

La interfaz gráfica está diseñada e implementada en Python con la librería Pyqt5, cuenta con todos los botones explicados en el apartado "INTERACCIONES CON EL USUARIO" (Figura 10).

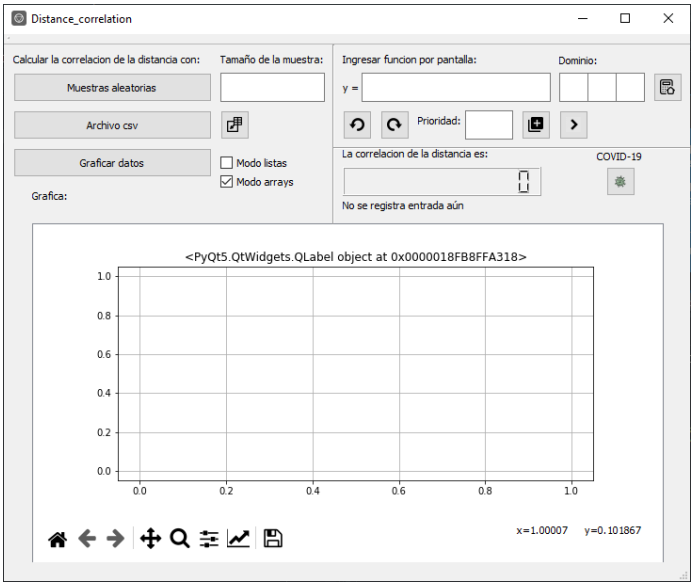


Figura 10

Por otra parte, la estructura del proyecto en GitHub (Figura 11) se presenta a continuación, donde se puede ver como se organizó la aplicación.

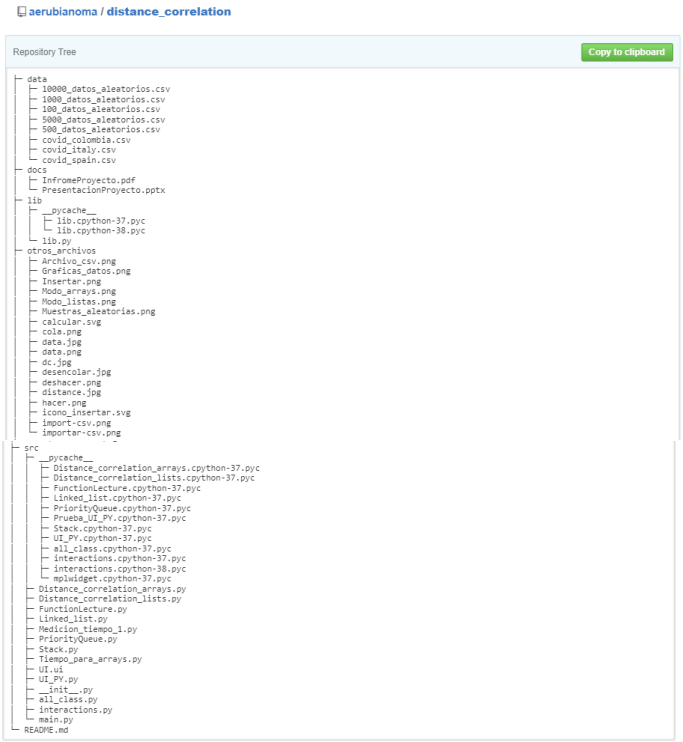


Figura 11

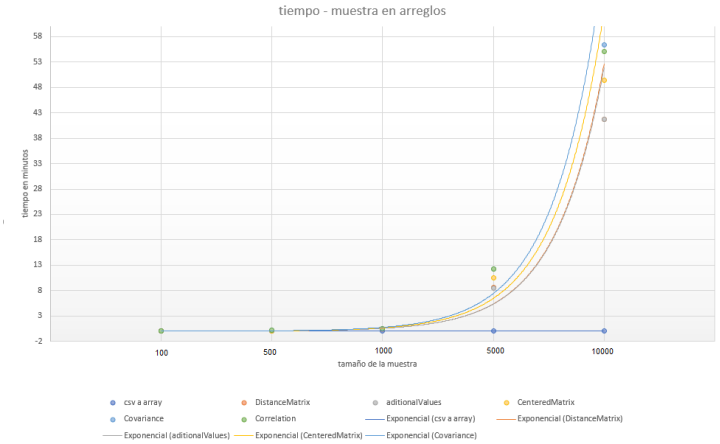
ENTORNO DE DESARROLLO DEL SISTEMA

El sistema está soportado en computador con sistema operativo Windows 8.1 pro, desarrollado en Python con el entorno de desarrollo de Anaconda – Spyder - Jupyter, con librerías como Pandas, Numpy, Pyqt5 y Timeit. Por su parte, los requisitos del computador en donde se ejecuta el programa son un procesador AMD FX(tm) – 6300 Six –Core 3.5GHz, memoria RAM 8GB ddr3, disco duro SanDisk SDSSDA120G ATA Device y GPU AMD Radeon R9 200 series.

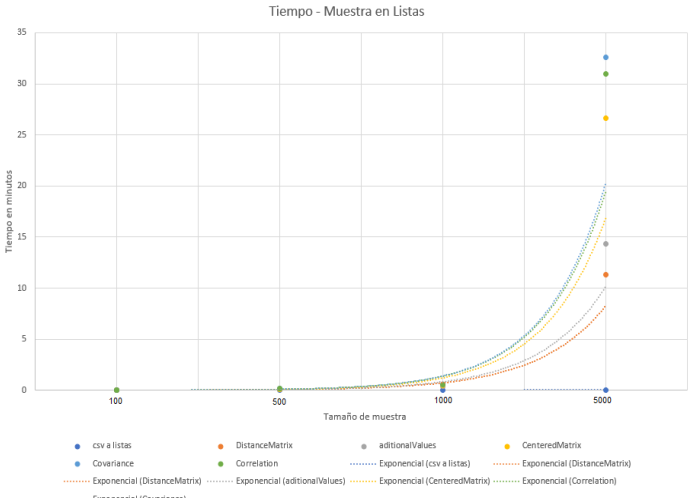
PRUEBAS DEL PROTOTIPO

Las pruebas realizadas al sistema se hicieron con las funcionalidades ingresar datos por medio de la librería Pandas, llenar matrices distantes, generar valores adicionales, llenar matrices centrales, calcular la distancia de covarianza, con muestras de 100, 500, 1000, 5000 y 10000 datos arrojando sus respectivos tiempos.

En los siguientes gráficos se puede apreciar el comportamiento que tiene el tiempo respecto al tamaño de muestra con las diferentes estructuras de datos (Grafica 1, Grafica 2).



Grafica 1

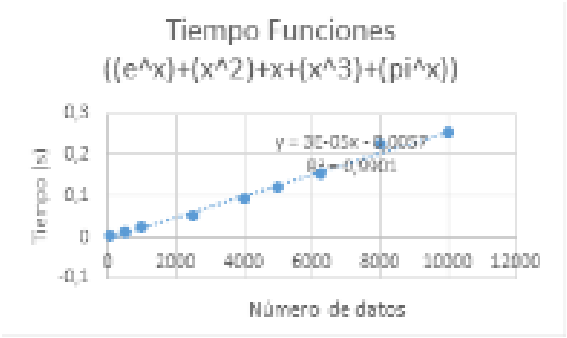


Grafica 2

Las pruebas realizadas a los arboles del prototipo se hicieron tomando como ejemplo una función cuadrática y otra polinómica junto con una exponencial, utilizando los datos con colas prioritarias e ingresándolos a través de la librería pandas. Se tomaron diferentes tamaños de muestra entre 2000 y 12000 datos arrojando sus respectivos tiempos los cuales se analizaron

dando como resultado un comportamiento de crecimiento lineal.
En los siguientes gráficos se puede apreciar el comportamiento que tiene el tiempo respecto al tamaño de muestra. (Grafica 3, Grafica 4).

Grafica 3



Grafica 4

Para las otras estructuras de datos no se realizaron pruebas ya que las operaciones que se realizaban con estas se realizan en un tiempo constante (Pilas, Colas, Hash)

ANÁLISIS COMPARATIVO

Por su parte en las siguientes tablas (Tabla 1, Tabla 2) y gráficos comparativos (Grafica 3, Grafica 4) se puede ver, cuáles fueron los datos obtenidos para las diferentes funcionalidades en ambos métodos de ingreso (listas y arrays).
Para los gráficos se tomó cada una de las barras de frecuencia como una funcionalidad distinta, en el eje x al tamaño de muestra y en el eje y el tiempo de ejecución.

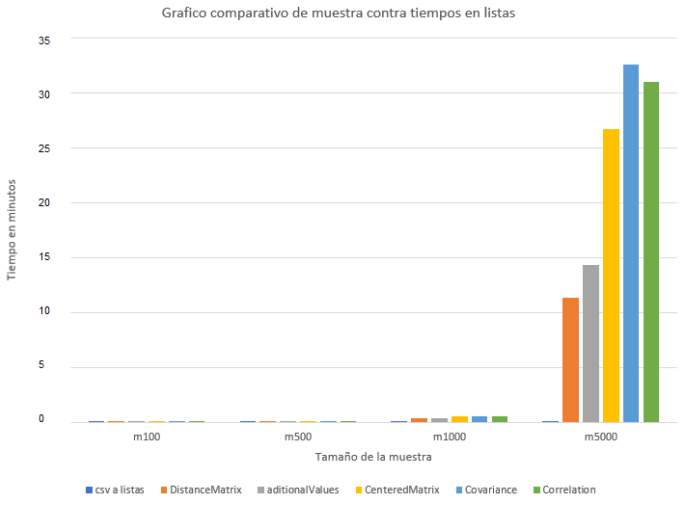
Tiempos en minutos para las funcionalidades con arreglos					
Tamaño muestra	100	500	1000	5000	10000
csv a array	0,0019	0,0048	0,0093	0,0461	0,0923
DistanceMatrix	0,0046	0,0874	0,3456	8,5594	41,7352
aditionalValues	0,0047	0,0886	0,3421	8,4788	41,6590
CenteredMatrix	0,0056	0,1060	0,4119	10,5402	49,4396
Covariance	0,0061	0,1218	0,4782	12,1982	56,3443
Correlation	0,0063	0,1219	0,4768	12,3111	55,1157

Tabla 1

Grafica 5

Tiempo en minutos para las funcionalidades				
Tamaño muestra	100	500	1000	5000
csv a listas	0,0016	0,0052	0,0098	0,0467
DistanceMatrix	0,0056	0,0980	0,3783	11,3474
aditionalValues	0,0054	0,1045	0,4084	14,3383
CenteredMatrix	0,0065	0,1303	0,5091	26,6442
Covariance	0,0070	0,1438	0,5721	32,6033
Correlation	0,0074	0,1414	0,5811	30,9743

Tabla 2



Grafica 6

Como se puede notar en las tablas y gráficos anteriores el crecimiento de los tiempos tiene un comportamiento O(n^2) puesto que al aumentar el tamaño de la muestra su tiempo respectivo se aproxima a una función cuadrática exceptuando la funcionalidad de ingreso de datos y está por su parte tiene un comportamiento O(n).

Como se observa en las gráficas la mejor opción para realizar el cálculo de la correlación de la distancia siempre será mejor con listas, optimizando así si el proceso.

Por otro lado, en las siguientes graficas 3 y 4 se observan cuáles fueron los tiempos obtenidos para las funciones con las cuales se realizaron las pruebas de tiempo

Para los gráficos se tomó como eje de las abscisas la cantidad de datos que se ingresaron como dominio de la función y como el eje de las ordenadas el tiempo de ejecución que tarda el programa. Como se puede apreciar en las gráficas el crecimiento del tiempo tiene un comportamiento lineal O(n) puesto que al aumentar el tamaño de los datos o tamaño de muestra su tiempo respectivo se aproxima a una función lineal.

ROLES Y ACTIVIDADES

Para la realización de este proyecto, Andrés Rubiano tomó los roles de líder y coordinador con su compromiso de programar y agendar reuniones por Google Meet y la mayoría del aporte teórico-práctico al proyecto. Por otra parte, Santiago Mariño tomo los roles de investigador y coordinador por su aporte a la consulta e investigación de las funciones y algoritmos para tomar el tiempo de las pruebas y Jorge Camacho tomo los roles de secretario y técnico al crear los informes y documentos del proyecto, además hacer la ejecución de los tiempos y toma de datos de las pruebas.

DIFICULTADES Y LECCIONES APRENDIDAS

- Implementación del cálculo de la *correlación de la distancia* con listas y arreglos.
- Creación de la interfaz gráfica.
- Creación del GitHub.
- No poder reunirse con los demás integrantes.
- Implementación de las técnicas de medición del tiempo para los distintos fragmentos del programa.
- Realizar las pruebas de los tiempos por falta de capacidad de computo.
- Adaptación al lenguaje de programación.

Las principales lecciones aprendidas son la capacidad de trabajo autónomo que se puede adquirir con herramientas como Google Meet, la diferencia notoria que se puede obtener al implementar estructuras de datos distintas, ver la utilidad que tiene un código organizado, la facilidad de utilizar GitHub para compartir código y archivos, el aprendizaje de nuevos algoritmos de búsqueda y ordenamiento que se deben tener para un código.