

Algoritmo de aceptación escolar - Tacna, Perú

Esta es una implementación del algoritmo de Gale-Shapley (algoritmo de aceptación diferida) para la asignación vacantes en instituciones educativas en Tacna, Perú.

Esta versión cuenta con tres posibles modos de funcionamiento, para entenderlos, es necesario introducir el concepto de preferencia imputada por distancia dentro del proceso de asignación:

Preferencia imputada por distancia

En el proceso regular de asignación, un postulante selecciona las instituciones por las que tiene preferencia y les asigna un orden de postulación. Sin embargo, debido a que cada estudiante postula a una cantidad limitada de instituciones, corre el riesgo de no alcanzar una vacante dentro de sus preferencias y quedar sin asignación. Para evitar estudiantes sin asignación, se realiza una imputación de prioridad por distancia. Es decir, para las instituciones que el estudiante no incluyó en su lista de prioridad, se calcula la distancia desde el punto de referencia del estudiante hasta cada una de las instituciones educativas (solo las que ofrecen el grado al que el estudiante postula). Se hace un ordenamiento de las instituciones educativas según distancia y se anexan a la lista de postulaciones del estudiante. Es decir, si el estudiante había postulado a 5 instituciones por preferencia, después de la imputación por distancia, el estudiante va a registrar postulaciones a todas las instituciones que ofrecen el grado de interés. Estos nuevos colegios se añaden en orden de preferencia consiguiente (en este ejemplo, la institución más cercana correspondería al orden de preferencia 6, y de manera consecutiva hasta N colegios, siendo N el orden de preferencia del colegio más lejano). Para identificar los colegios que han sido imputados por distancia, es decir, que no fueron elegidos por el estudiante, se crea una variable "distancePriority" en la tabla de postulaciones (demand.csv), que indica verdadero si la postulación de un estudiante a una institución ha sido imputada por distancia y falso en caso contrario.

A partir de este concepto de preferencia por distancia, se crean los tres modos de funcionamiento que ofrece el algoritmo:

1. Asignación sin preferencia por distancia:

El proceso de asignación escolar toma en cuenta únicamente las preferencias establecidas por el postulante. Si las instituciones seleccionadas ocupan todas sus vacantes, el estudiante no queda asignado a ningún establecimiento.

2. Asignación con preferencia por distancia previamente calculada:

En este caso, se asume que el archivo con las postulaciones (demand.csv) contiene postulaciones imputadas por distancia. Es decir, cada estudiante postula a todas las instituciones posibles y existe una columna adicional llamada "distancePriority" que indica verdadero para las postulaciones que han sido imputadas por distancia.

Si se ejecuta el algoritmo bajo esta modalidad y el archivo "demand.csv" no cuenta con la columna "distancePriority", el algoritmo va a mostrar un mensaje de error.

3. Asignación con preferencia por distancia, cálculo realizado por el algoritmo:

Este escenario corresponde al caso en que se desea incluir la imputación de preferencia por distancia, pero únicamente se cuenta con las postulaciones elegidas por los estudiantes. En este caso, el algoritmo está en capacidad de tomar la ubicación (latitud, longitud) de cada uno de los estudiantes y utilizarla para calcular la distancia a cada institución. El resultado es que se va a generar un nuevo archivo de postulaciones (demand.csv) que va a incluir las postulaciones imputadas para cada estudiante según orden de distancia. También se va a agregar la columna "distancePriority" de manera automática. Una vez hecho esto, el algoritmo toma este nuevo archivo y ejecuta la asignación con preferencia por distancia.

Es requisito que, para este modo de uso, la tabla de postulantes (postulants.csv) y la tabla de vacantes de las instituciones educativas (vacancies.csv) contengan la ubicación geográfica (latitud y longitud). Estos campos son luego tomados para hacer el cálculo de distancia.

Nota: el cálculo que se realiza corresponde a una distancia lineal y no una distancia de viaje por carretera.

Descripción de inputs

Según el modo elegido para ejecutar el algoritmo, las tablas de entrada pueden variar. A continuación se describen los campos mínimos necesarios para la ejecución de este (otros campos pueden aparecer, pero no son tenidos en cuenta).

Es importante notar que, para poder reconocer las tablas, los nombres deben mantenerse como se muestra a continuación.

- **Base de datos de vacantes:** vacancies.csv

- localId
- serviceId
- annex
- areaId
- studentBodyId
- levelId
- gradeId
- shiftId
- studentModalityId
- classroomTypeId
- totalVacancyNna
- totalVacancyNnaNee
- roundNumber
- roundTypeId
- sendDate

Adicionalmente, si se desea que el algoritmo haga el cálculo de distancia para la imputación de preferencia por distancia, esta tabla debe contener también:

- latitude
- longitude
- **Base de datos de demanda:** demand.csv
 - postulantId
 - levelId
 - gradeId
 - order
 - serviceId
 - localId
 - priority
 - roundNumber
 - roundTypeId
 - sendDate
- **Base de datos de identificación de estudiantes:** postulants.csv
 - postulantId
 - priority
 - guardianId
 - roundNumber
 - roundTypeId
 - sendDate

Adicionalmente, si se desea que el algoritmo haga el cálculo de distancia para la imputación de preferencia por distancia, esta tabla debe contener también:

- latitude
- longitude
- **Base de datos de solicitudes grupales:** postulations.csv
 - postulantId
 - guardianId

- `typeld`
- `roundNumber`
- `roundTypeld`
- `sendDate`

Instrucciones para configurar el ambiente de ejecución.

1. Tener instalada alguna distribución de Python 3.6 o superior.
2. Abrir el Terminal y asegurarse de tener el comando `python` como variable en la terminal. Esto se chequea corriendo el comando `python --version`.
3. En caso de que ser usuario de Windows, utilizar **Windows PowerShell** (no utilizar **Command Prompt**). Para que el comando funcione se deberá seguir [estas instrucciones](#).
4. Actualizar `pip`:
 - Mac/Linux: `python -m pip install --user --upgrade pip`
 - Windows: `python -m pip install --upgrade pip`. Si no funciona, probar reemplazando `python` con `py`
5. Instalar `virtualenv`.
 - Mac/Linux: `python -m pip install --user virtualenv`
 - Windows: `python -m pip install --user virtualenv`. Si no funciona, probar reemplazando `python` con `py`
6. Una vez que ya se tenga Python instalado en el computador (chequeando que `python --version` entrega un número de versión), moverse al directorio en donde se tiene la carpeta con el algoritmo (carpeta de nombre `cb-da`, la misma carpeta donde está ubicado este archivo README.md). Esto se hace con el comando `cd`.
7. Una vez en la carpeta, crear un ambiente virtual utilizando `python -m venv .` (incluir el punto al final del comando, eso indica que se creará un entorno de trabajo en la carpeta actual). Para activarlo:
 - Mac/Linux: `source /bin/activate`
 - Windows: `Scripts\activate`. Si no funciona, [este post](#) puede ser útil.
8. El ambiente virtual estará correctamente activado si aparece el nombre del folder entre paréntesis a la izquierda de la línea de comandos. Siempre que se quiera correr el algoritmo, hay que estar seguro de que este ambiente este activado.
9. Instala los requerimientos utilizando `pip install -r requirements.txt`.

Ejecución del algoritmo

El algoritmo se ejecuta con **archivos de configuración**. Estos archivos indican al algoritmo cuáles son los parámetros con los que debe llevarse a cabo la asignación. Adicionalmente, en este archivo debe especificarse la ruta a la carpeta que contiene los archivos de entrada para poder ser leídos por el código.

Cada modo de funcionamiento tiene un archivo de configuración diferente que le indica al algoritmo qué tipo de asignación ejecutar así:

1. Asignación sin preferencia por distancia:

Para ejecutar el algoritmo en este modo, se utiliza el archivo de configuración llamado: `da_tacna_sin_distancia.py`.

Este archivo se encuentra dentro del folder "cb_da". Antes de llevar a cabo la ejecución, debe abrir este archivo e insertar la ruta al folder que contiene los archivos de entrada. Para hacer esto, en el archivo `da_tacna_sin_distancia.py` se crea una variable llamada "dir", esta variable debe cambiarse por la ruta en su computador (en algunos casos puede presentarse error en la lectura si no se incluye el separador "/" al final de la ruta. Asegúrate de utilizar esta notación).

Para correr el algoritmo, vaya a la terminal y ejecute el comando: `python cb_da/da_tacna_sin_distancia.py`

Los archivos producidos con las asignaciones se guardarán en la misma carpeta que contiene los archivos de entrada.

2. Asignación con preferencia por distancia previamente calculada:

Para ejecutar el algoritmo en este modo, se utiliza el archivo de configuración llamado: `da_tacna_distancia_precalculada.py`.

Este archivo se encuentra dentro del folder "cb_da". Antes de llevar a cabo la ejecución, debe abrir este archivo e insertar la ruta al folder que contiene los archivos de entrada. Para hacer esto, en el archivo `da_tacna_distancia_precalculada.py` se crea una variable llamada "dir", esta variable debe cambiarse por la ruta en su computador (en algunos casos puede presentarse error en la lectura si no se incluye el separador "/" al final de la ruta. Asegúrate de utilizar esta notación).

Para ejecutar este algoritmo de manera exitosa, recuerde que el archivo de postulaciones (`demand.csv`) debe contener una columna indicando cuáles postuaciones han sido imputadas por distancia.

Para correr el algoritmo, vaya a la terminal y ejecute el comando: `python cb_da/da_tacna_distancia_precalculada.py`

Los archivos producidos con las asignaciones se guardarán en la misma carpeta que contiene los archivos de entrada.

3. Asignación con preferencia por distancia, cálculo realizado por el algoritmo:

Para ejecutar el algoritmo en este modo, se utiliza el archivo de configuración llamado: `da_tacna_distancia_calculada_lineal.py`.

Este archivo se encuentra dentro del folder "cb_da". Antes de llevar a cabo la ejecución, debe abrir este archivo e insertar la ruta al folder que contiene los archivos de entrada. Para hacer esto, en el archivo `da_tacna_distancia_calculada_lineal.py` se crea una variable llamada "dir", esta variable debe cambiarse por la ruta en su computador (en algunos casos puede presentarse error en la lectura si no se incluye el separador "/" al final de la ruta. Asegúrate de utilizar esta notación).

Para ejecutar este algoritmo de manera exitosa, recuerde que los archivos `postulants.csv` y `vacancies.csv` deben contener información con la latitud y longitud de las ubicaciones geográficas respectivamente, ya que se hace un cálculo de distancia.

En este caso se crea el archivo de demanda con preferencia imputada por distancia. Este procedimiento puede tardar varios minutos debido al alto consumo de recursos de procesamiento. Sin embargo, el archivo producido "`demand_with_distance.csv`" se guarda en el mismo folder que contiene los archivos de entrada. Este archivo "`demand_with_distance.csv`" puede usarse posteriormente para ejecutar el algoritmo bajo el modo de funcionamiento 2 (distancia precalculada). Así, no es necesario esperar nuevamente a que el algoritmo produzca este archivo.

Para correr el algoritmo, vaya a la terminal y ejecute el comando: `python cb_da/da_tacna_distancia_calculada_lineal.py`

Los archivos producidos con las asignaciones se guardarán en la misma carpeta que contiene los archivos de entrada.

Archivos producidos por el algoritmo.

El algoritmo produce dos archivos de salida que se encuentran en el mismo folder que contiene los archivos de entrada:

- `asignaciones.csv`: Indica los estudiantes y los respectivos programas a los que fueron asignados
 - `postulantId`
 - `localId`
 - `serviceId`
 - `annex`
 - `studentBodyId`
 - `levelId`
 - `gradId`
 - `shiftId`
 - `studentModalityId`
 - `classroomTypeId`
 - `assigned`
 - `assignmentTypeId`

- roundNumber
- roundTypeId
- sendDateTime
- **lottery_numbers.csv**: Indica el número de lotería asignado a cada postulante por cada uno de los programas seleccionados.
 - postulantId
 - order
 - lottery_number_quota