

Seminario Redes y Voto

Juan Manuel Aneas Franco



**UNIVERSIDAD
DE GRANADA**

Índice

Variables Estudio	3
Funcionamiento Inicial del Modelo	3
Archivo Sin_Red.es.py	4
Archivos Facebook.py, Twitter.py e Instagram.py	5
Conclusión	5

Variables Estudio

Estas son las variables que se manejan a lo largo del estudio. Las dos últimas son generadas por el modelo. El resto las encontramos en el excel de Datos Útiles.

A1: Valoración Económica Sociotrópica Actual. Valores de 0 a 4.

A2: Valoración Económica Sociotrópica Retrospectiva. Valores de 0 a 2.

A3: Valoración Económica Sociotrópica Prospectiva. Valores de 0 a 2.

A4: Valoración Económica Egotrópica Actual. Valores de 0 a 4.

A5: Valoración Económica Egotrópica Prospectiva. Valores de 0 a 2.

A6: Valoración Económica Política Actual. Valores de 0 a 4.

A7: Valoración Económica Política Retrospectiva. Valores de 0 a 2.

A8: Valoración Económica Política Prospectiva. Valores de 0 a 2.

B10_1: Uso de Facebook. Valores posibles: 1 o NULL.

B10_2: Uso de Twitter. Valores posibles: 1 o NULL.

B10_3: Uso de Instagram. Valores posibles: 1 o NULL.

B11_5: Uso de Redes Sociales para seguir la campaña. Valores de 0 (no) a 1 (sí).

C7A: Voto. Lo transformamos al valor 1 si se vota al partido del Gobierno. 0 en caso contrario.

C7: El votante ha votado o no. Únicamente nos quedamos con los votantes que tienen valor 1 en esta variable, cuando el votante fue a votar y votó.

DatosEstimados: Valor estimado por el modelo para cada votante (0 en caso de no votar al partido el Gobierno y 1 en caso contrario).

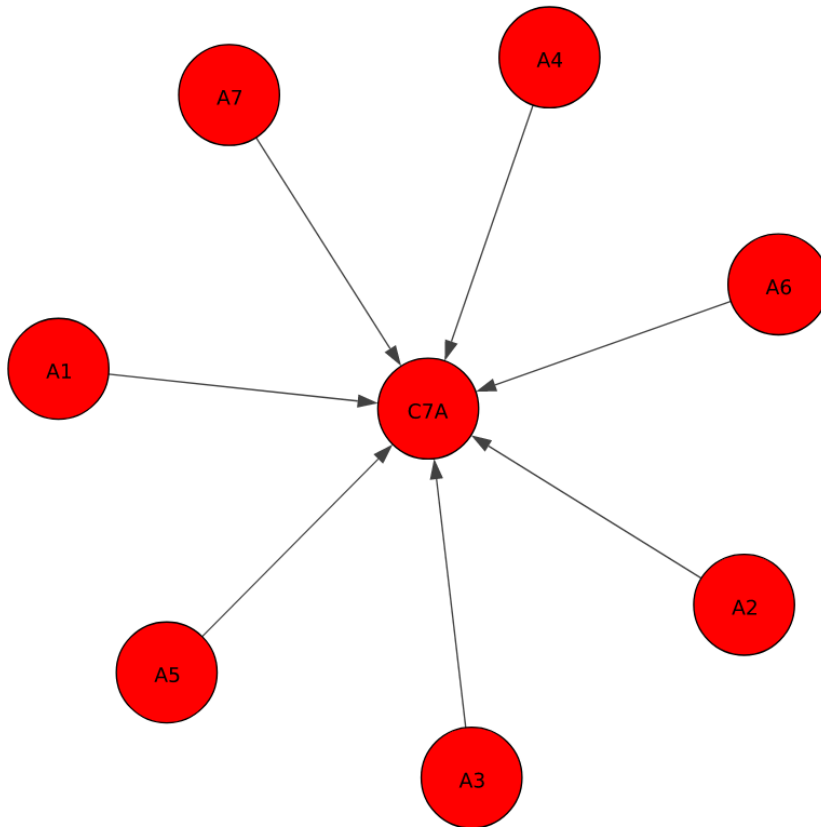
DatosTrasRed: Valor estimado por el modelo para cada votante después de aplicar el efecto red si el votante usa redes para seguir la campaña (0 en caso de no votar al partido el Gobierno y 1 en caso contrario).

Funcionamiento Inicial del Modelo

En primer lugar debemos ejecutar el archivo main.py, archivo principal que llamará al resto. En la primera parte de este archivo, se indican las rutas en las que se va a generar cada excel. A continuación, se define un método llamado excel_estudio() que será el encargado de crear un excel por cada ruta establecida anteriormente partiendo del excel de Datos Útiles. ¿Por qué se hace esto? Para poder separar cada análisis. Por un lado, el análisis de los votantes sin tener en cuenta si usan redes sociales o no. Por otro lado, un análisis distinto para los votantes que usan Facebook, Twitter o Instagram. Este método filtra los datos del excel de Datos Útiles y asigna a cada uno de los nuevos excel sus correspondientes datos. Emplea los métodos transformar_voto() y normalizar() del archivo metodos.py. En ese archivo aparece explicado en comentario su funcionalidad. Finalmente, este archivo despliega un menú mediante interfaz gráfica que permite elegir uno de los 4 estudios posibles: sin redes, facebook, twitter e instagram. A continuación se explica cada uno de ellos.

Archivo Sin_Red.es.py

El archivo `sin_redes.py` corresponde con el estudio más simple, ya que parte de un excel en el que se encuentran los datos de todos los votantes que fueron a votar y votaron, independientemente de si usaron redes para seguir la campaña o no. En nuestro modelo, la variable `C7A` queda determinada por las otras 8 variables (`A1-A8`).



¿En qué momento establecemos esta red en el código? En la inferencia aprovechando la librería `PyAgrum` de Python. A través de esta librería podemos generar un learner que carga todos los datos del excel correspondiente. A partir de ese learner, podemos establecer una evidencia, que va a corresponder con las 8 variables que determinan `C7A` en nuestro modelo, que permita inferir una probabilidad para la variable a determinar, el Voto. Para cada votante, el modelo toma sus valores de `A1` a `A8` y estima un valor para el voto (0 o 1). Este es el valor que se añade a la columna `DatosEstimados`, uno por cada votante. Finalmente mostramos el error cuadrático medio comparando la columna `DatosEstimados` con la columna `C7A` (voto real).

Archivos Facebook.py, Twitter.py e Instagram.py

En estos archivos vamos un punto más allá, ya que corresponde con estudios/análisis en los que sí vamos a tener en cuenta que el votante usa redes sociales para seguir la campaña. Además, incluimos un efecto red para ver si el votante modifica su decisión. Los tres archivos son prácticamente iguales en código, solo que cada uno de ellos parte de su correspondiente excel (facebook.py trabaja sobre los votantes que han usado Facebook para seguir la campaña, idem para los demás). La explicación de cada línea de código se encuentra mediante comentarios en los archivos. Inicialmente estos tres archivos, al igual que el archivo sin_redes.py, generan la columna DatosEstimados, valores que estima el modelo para cada votante a través del learner de PyAgrum que carga todos los datos y establece como evidencia las variables A1-A8. Una vez tenemos la columna DatosEstimados, al igual que en el archivo sin_redes.py, mostramos el error cuadrático medio comparando con la columna C7A para comparar el dato con el resultado tras aplicar el efecto red.

A continuación, se aplica el efecto red empleando el método efecto_red() del archivo metodos.py. Aparece explicado en comentario, como todo el código en todos los archivos, pero el funcionamiento y la idea es tan simple como funcional, ya que nos permite mejorar el error cuadrático medio. ¿Qué se ha propuesto? Que cuando el DatoEstimado para un votante difiere del voto real, mediante una simulación se determine una probabilidad de cambio y, en caso de darse esa probabilidad, cambiar la decisión. Durante el desarrollo del programa, he establecido esa probabilidad en un 45%. A mayor probabilidad de cambio, es más posible cambiar la decisión y acercarse a la realidad, pero no sería un modelo lógico. Todo depende de cada persona, pero me he decantado por elegir que un votante priorice en mayor parte su idea inicial, por lo que el cambio de decisión se da en menos casos. Dado que no podemos evaluar cuántos miembros hay en la red, ya que partimos de unos datos fijos y otros estimados, el efecto red ha consistido en cambiar la decisión de un votante un 45% de las veces si el modelo estima un dato distinto al real. Estos nuevos valores, para cada votante, se han incluido en la columna DatosTrasRed. Finalmente volvemos a mostrar el error cuadrático medio, pero esta vez comparando las columnas DatosTrasRed y C7A. Obtenemos un valor menor, ya que hay valores que han variado aproximándose más a la realidad.

Conclusión

Este proyecto ha sido fruto del análisis de mucha de la información disponible. Ha resultado mucho más fácil la implementación de lo pensado gracias a la librería PyAgrum que se exponía en la guía del proyecto. Esta librería ha permitido no tener que implementar las distribuciones probabilísticas e inferencias que proponía Juan Sosa ya que tiene métodos con una eficiencia y eficacia similar. Sí, parece que Python tiene librerías para todo.

Por otro lado, este modelo podría mejorarse, por ejemplo, añadiendo más variables que determinen la variable voto. Para poder hacer algo funcional, partí únicamente de las 8 variables A1-A8 para determinar el voto, y finalmente no encontré el momento para añadir alguna más, como por ejemplo tener en cuenta la afinidad ideológica de cada votante. También sería curioso tratar de redistribuir el peso de cada una de las variables, determinando alguna que otra en mayor parte a la variable estimada.

Finalmente se ha logrado un programa bien estructurado con un código bastante simple que se basa en una librería proporcionada por Python. Hemos obtenido ese resultado, una interfaz gráfica a través de la cuál podemos generar unos excel y ver cómo se estima un valor de voto para cada análisis mostrando el correspondiente error cuadrático medio. Se ha demostrado que el efecto red elegido ha ayudado a reducir el error mencionado anteriormente.