Universidad de Buenos Aires

Facultad de Ciencias Exactas y Naturales

Maestría en Explotación de Datos y Descubrimiento del Conocimiento

Trabajo de especialización

Julio 2020

Explicaciones contrafácticas de modelos de aprendizaje automático complejos

Autor: **Lucas Pecina**

Primera presentación - AVANCE DEL TRABAJO

29/07/2020

Para ésta primera presentación del trabajo de especialización definí la estructura principal de lo que va a ser el trabajo para tener una idea de cual es el producto final.

En primer lugar se hace una introducción teórica a la problemática a resolver, el método usado para ello y distintas variantes similares.

Luego, se presenta un caso sencillo de uso: hago una breve descripción del dataset simple usado como ejemplo de juguete (PISA USA 2009), cuento su pre-procesamiento y también el modelado y entrenamiento de una red neuronal para clasificación. Una vez entrenado y testeado, también se da muestra de las variantes contrafácticas “hechas a mano”, o sea, sin ninguna librería de python especializada. Por último se visualizan los resultados y se muestra como para dos instancias distintas, las explicaciones contrafácticas recomiendan diferentes situaciones (esa es la importancia de estas técnicas).

El siguiente paso es explicar un problema de implementación que estos métodos “a mano” tienen y presentar una solución basada en un paper donde se lo trata como un problema de optimización. También se presenta una librería especializada en explicaciones contrafácticas y se hacen pruebas sobre el mismo dataset de juguete. Se muestran sus resultados.

Luego, se pasa a probar estas técnicas en un dataset más complejo, de la misma temática que el anterior pero con muchas más variables (PISA 2009 extendido). Se analiza el preprocesamiento realizado, el entrenamiento de la red neuronal para clasificación y los resultados arrojados. Se estudian los contrafácticos propuestos para diversos individuos. Por último se aplica a un nuevo dataset (de ataques al corazón) con el mismo procedimiento que los anteriores.

Se termina con un conjunto de conclusiones y posibilidades para ampliar el trabajo realizado.

ACLARACIÓN: para esta presentación se muestran las conclusiones y resultados que se tienen hasta el momento. Sigo trabajando en pulir muchas cosas y falta también extender la teoría. Aclararé en cada parte las cosas que faltan ser terminadas o mejoradas.

Índice general

1. Resumen

El presente trabajo consiste en una introducción al tema de las **explicaciones contrafácticas de modelos estadísticos de aprendizaje automático**, poniendo énfasis en la problemática a resolver, la importancia y utilidad de estas técnicas y su aplicación en el mundo real.

Se hará énfasis en demostrar que las explicaciones contrafácticas pueden ser utilizadas como un complemento post-hoc de los modelos complejos de machine learning en dos grandes frentes: **la toma de decisiones a nivel individual** y la **interpretabilidad del modelo** para el análisis de sesgos.

Por último se analizará su aplicabilidad a casos puntuales de la vida real, sus limitaciones y potenciales mejoras a estos métodos.

2. Introducción

2.1. Contexto y problema a resolver

Existe un clasico trade-off entre la interpretabilidad y el poder predictivo de los modelos estadísticos según su complejidad. Si optamos por utilizar modelos simples, es más probable que logremos entender la influencia de las variables en el resultado final. Esto nos permite inferir ciertas propiedades del mundo real cuando queremos hacer investigación científica o tomar decisiones basadas en dichas variables.

Sin embargo, la simpleza de estos modelos hace que no logren captar mucha de la complejidad que tiene el mundo real, perjudicando así el poder predictivo del modelo.

Cuando se requiere predecir con la mayor exactitud posible, se utilizan modelos complejos, los llamados black-boxes. Estos logran un mejor rendimiento pero a costas de su falta de transparencia. Por lo que si lo único que nos interesa es obtener la probabilidad final de un resultado, los black-box son el camino a seguir.

Pero qué pasa si se nos presenta un escenario en el cual debemos hacer predicciones correctas y a su vez entender el motivo por el cual el modelo predice dicho resultado a nivel individual? Por ejemplo si una persona aplica para un préstamo y es rechazado por el algoritmo del banco, queremos entender por que el usuario fue rechazado y que podría haber hecho distinto

para ser aprobado. Esto podría ayudar a dar recomendaciones que sirvan para informar la toma de decisiones.

En el caso de usar modelos complejos, realizaremos buenas predicciones pero a costa de no saber los motivos que llevaron a ese resultado. Si usamos modelos simples, no solo que las predicciones son menos confiables, sino que también encontramos otro problema: la falta de complejidad no va a permitir captar las interacciones y no linealidades que ocurren a nivel individual. Estos modelos nos pueden decir que, **en el agregado**, una variable está relacionada positivamente con el resultado, sin embargo, no nos dirá cómo esa variable se relaciona con el resultado para cierto **individuo particular**.

Por eso es necesario tener un método para tomar lo mejor de los dos enfoques: aproximar una función lo suficientemente compleja para tener un buen poder de predicción y, a su vez, poder apreciar los efectos de las variables en el resultado. Todo esto a nivel del individuo. Las **explicaciones contrafácticas** pueden ayudar a solucionar este problema.

2.2. Explicaciones contrafácticas

Una explicación contrafáctica (ECF) describe una situación causal de la forma "Si la situación X hubiera sido diferente, el evento Y habría ocurrido?". La idea general es que las ECF pueden ser usadas para explicar predicciones de modelos de aprendizaje automático complejos en instancias individuales. En este caso, el "evento" Y es el resultado predicho por el modelo dada la "situación" X, que es el conjunto de valores que toman las variables en una instancia particular.

Cuando el modelo es un black-box, partimos de asumir que todas las variables causan la predicción (independientemente de si hay una relación causal en la vida real entre ellos). Luego, elegimos una instancia inicial de interés y simulamos distintos escenarios contrafácticos a partir de esa instancia, con el objetivo de analizar cómo cambia el resultado predicho ante perturbaciones de estas variables.

Esto nos puede servir para cumplir con dos objetivos diferentes:

- Descriptivo: ver el efecto de perturbar ciertas variables mientras las demás permanecen fijas y observar el efecto en el resultado a nivel individual. Puede servir para analizar sesgos en el modelo.
- Decisiones: buscar en el espacio de contrafácticos hasta obtener el resultado predicho que nos interesa y ahí observar los cambios mínimos necesarios para llegar a ese resultado.

2.3. Aplicaciones

Volviendo al ejemplo anterior del préstamo bancario, podemos tomar este doble enfoque para analizar distintos aspectos del modelado.

Por un lado, podemos preguntarnos cosas del estilo: si el individuo aplicando al credito hubiera sido de la raza R2 en lugar de R1, le habrian dado el credito? Y si hubiera sido del sexo S2 en vez de S1? Este tipo de preguntas nos puede ayudar a identificar los sesgos que puede tener el modelo para hacerlos mas justos.

Por otro lado, es muy útil saber qué es lo mínimo que podría haber sido diferente del individuo que aplicó al crédito, para que el modelo se lo otorgara. Entonces, exploramos el espacio de contrafácticos hasta identificar el conjunto de cambios en las variables que hace esto posible. De esta manera, el modelo puede recomendar que decisiones tomar.

FALTA PULIR ALGUNAS CUESTIONES DE REDACCIÓN Y AGREGAR UN POCO MAS DE INFO

3. Ejemplo ilustrativo: modelo simple

A continuación se utilizará un dataset simple (de pocas variables) para realizar todo el proceso de clasificación y posterior explicación contrafáctica para sacar algunas conclusiones. Para esto, se analizará el dataset, se construirá una red neuronal para clasificación y, luego, se crearán “a mano” ejemplos contrafácticos partiendo de distintas instancias.

3.1. Dataset PISA USA 2009

El dataset “de juguete” elegido para realizar estas primeros análisis post-hoc es el de resultados de pruebas PISA en literatura para los Estados Unidos en el año 2009.

Link al dataset: https://www.kaggle.com/econdata/pisa-test-scores

Contexto:

El informe del programa internacional para la Evaluación de Estudiantes o Informe PISA es un estudio llevado a cabo por la OCDE a nivel mundial que mide el rendimiento académico de los alumnos en matemáticas, ciencia y lectura. Su objetivo es proporcionar datos comparables que posibiliten a los países mejorar sus políticas de educación y sus resultados.
El estudio se basa en el análisis del rendimiento de estudiantes de 15 años a partir de unos

exámenes estandarizados que, desde el año 2000, se realizan cada tres años en diversos países.

Dataset:

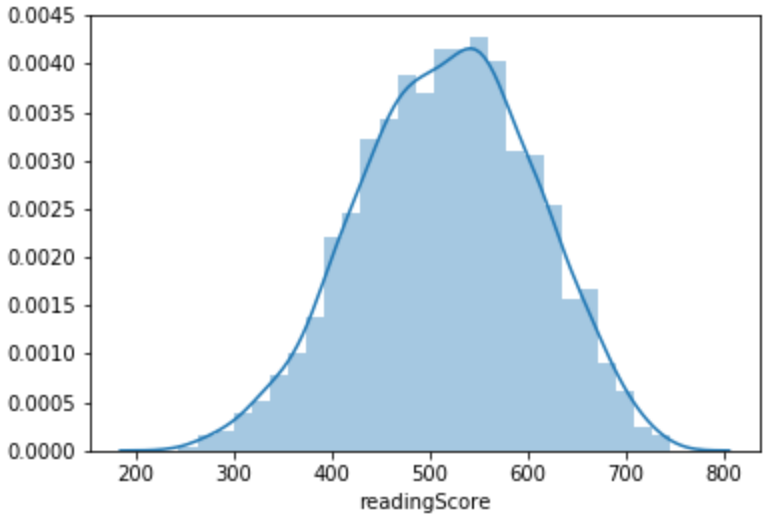
Este conjunto de datos está armado para predecir el resultado de las pruebas de lectura de estudiantes de los Estados Unidos en el año 2009.
Las variables con las que se cuenta contienen información acerca de la demográfica y escuelas de los estudiantes, derivado de los Archivos de Datos de Uso Público del Centro de Estadísticas de Educacion de los Estados Unidos (NCES). Cada fila representa a un estudiante que toma el examen, hay 5232 en total.

Las variables son:

- grade: The grade in school of the student (most 15-year-olds in America are in 10th grade)
- male: Whether the student is male (1/0)
- raceeth: The race/ethnicity composite of the student
- preschool: Whether the student attended preschool (1/0)
- expectBachelors: Whether the student expects to obtain a bachelor's degree (1/0)
- motherHS: Whether the student's mother completed high school (1/0)
- motherBachelors: Whether the student's mother obtained a bachelor's degree (1/0)
- motherWork: Whether the student's mother has part-time or full-time work (1/0)
- fatherHS: Whether the student's father completed high school (1/0)
- fatherBachelors: Whether the student's father obtained a bachelor's degree (1/0)
- fatherWork: Whether the student's father has part-time or full-time work (1/0)
- selfBornUS: Whether the student was born in the United States of America (1/0)
- motherBornUS: Whether the student's mother was born in the United States of America (1/0)
- fatherBornUS: Whether the student's father was born in the United States of America (1/0)
- englishAtHome: Whether the student speaks English at home (1/0)
- computerForSchoolwork: Whether the student has access to a computer for schoolwork (1/0)
- read30MinsADay: Whether the student reads for pleasure for 30 minutes/day (1/0)
- minutesPerWeekEnglish: The number of minutes per week the student spend in English class
- studentsInEnglish: The number of students in this student's English class at school
- schoolHasLibrary: Whether this student's school has a library (1/0)
- publicSchool: Whether this student attends a public school (1/0)
- urban: Whether this student's school is in an urban area (1/0)
- schoolSize: The number of students in this student's school

Target: readingScore: The student's reading score, on a 1000-point scale.

Distribución del target:



Preprocesamiento:

Para lograr obtener un dataset en condiciones de ser utilizado en un modelo de clasificación (en este caso una red neuronal), se debieron realizar los siguientes procesos:

1. En primer lugar se binariza el target con un punto de corte en 500 puntos. La idea es entonces predecir aquellos alumnos que superan los 500 puntos en el examen de lectura de PISA en Estados Unidos en 2009.

Distribución de clase en dataset de entrenamiento:

Clase	Cantidad
< 500	1834
> 500	1829

2. Luego, se quitan aquellas filas que contienen datos incompletos o NAs.

3. Se realiza un one-hot encoding a la variable “raceeth” que es la única categórica.

Modelo de clasificación:

Red neuronal:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                multiple                  870
_________________________________________________________________
dense_1 (Dense)              multiple                  1500
_________________________________________________________________
dense_2 (Dense)              multiple                  765
_________________________________________________________________
dense_3 (Dense)              multiple                  16
=================================================================
Total params: 3,151
```

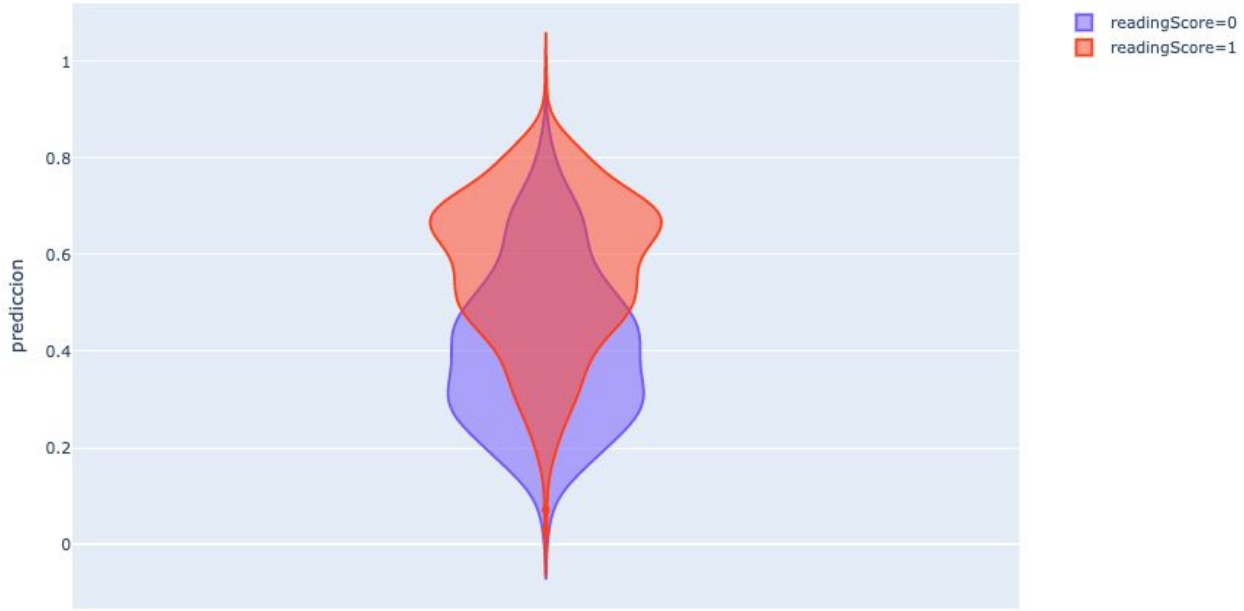
ARMAR MEJOR LAS TABLAS Y PONER GRÁFICOS DE ROC CURVE

Evaluacion:

Matriz de confusión:

288	115
172	415

Accuracy: 0.7101010101010101
f1: 0.7430617726051926
recall: 0.706984667802385
precision: 0.7830188679245284



VER SI DEJAR TODAS LAS VARIABLES, HACER SELECCIÓN O HACER SOLO EXPLICACION

3.2. Ejemplo contrafáctico

El primer intento de analizar contrafácticos consiste en la creación artificial y arbitraria de distintos escenarios (cambios en valores de ciertas variables) a partir de una instancia determinada también arbitrariamente. Con esto veremos como el modelo "recomienda" situaciones cercanas al individuo estudiado, con el fin de comparar esas recomendaciones con las hechas para otro individuo diferente.

Instancia (individuo) 1:

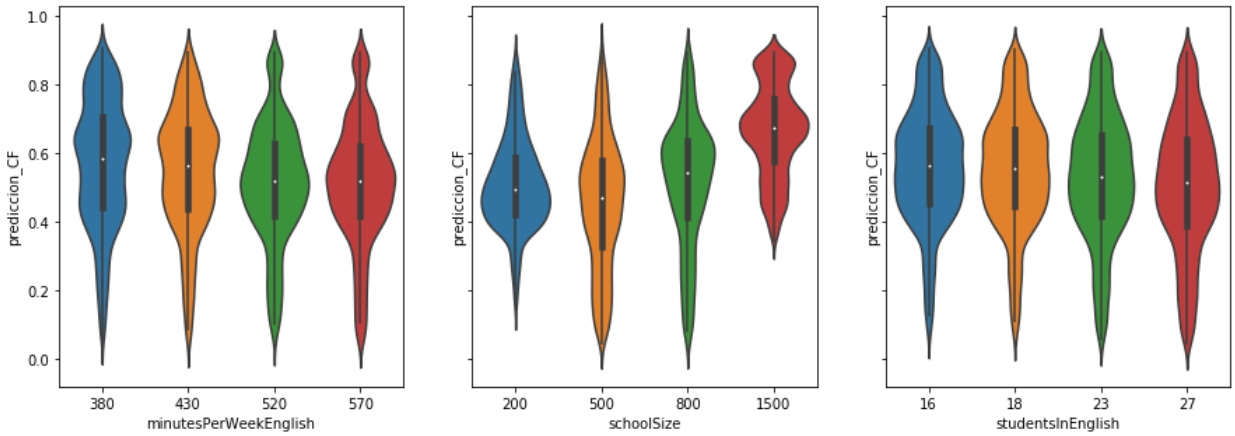
grade	10.0
male	0.0
preschool	0.0
expectBachelors	0.0
motherHS	1.0
motherBachelors	0.0
motherWork	0.0
fatherHS	0.0
fatherBachelors	0.0
fatherWork	1.0
selfBornUS	1.0
motherBornUS	1.0
fatherBornUS	1.0
englishAtHome	1.0
computerForSchoolwork	0.0
read30MinsADay	1.0
minutesPerWeekEnglish	480.0
studentsInEnglish	20.0
schoolHasLibrary	0.0
publicSchool	1.0
urban	1.0
schoolSize	626.0
raceeth_American Indian/Alaska Native	0.0
raceeth_Asian	0.0
raceeth_Black	0.0
raceeth_Hispanic	0.0
raceeth_More than one race	0.0
raceeth_Native Hawaiian/Other Pacific Islander	0.0
raceeth_White	1.0

Las variables a perturbar son las siguientes: **'expectBachelors', 'motherWork', 'fatherWork', 'computerForSchoolwork', 'read30MinsADay', 'minutesPerWeekEnglish', 'studentsInEnglish', 'schoolHasLibrary', 'publicSchool', 'urban', 'schoolSize'.** Hay que recordar que las perturbaciones se hacen de forma arbitraria.

Resultados:

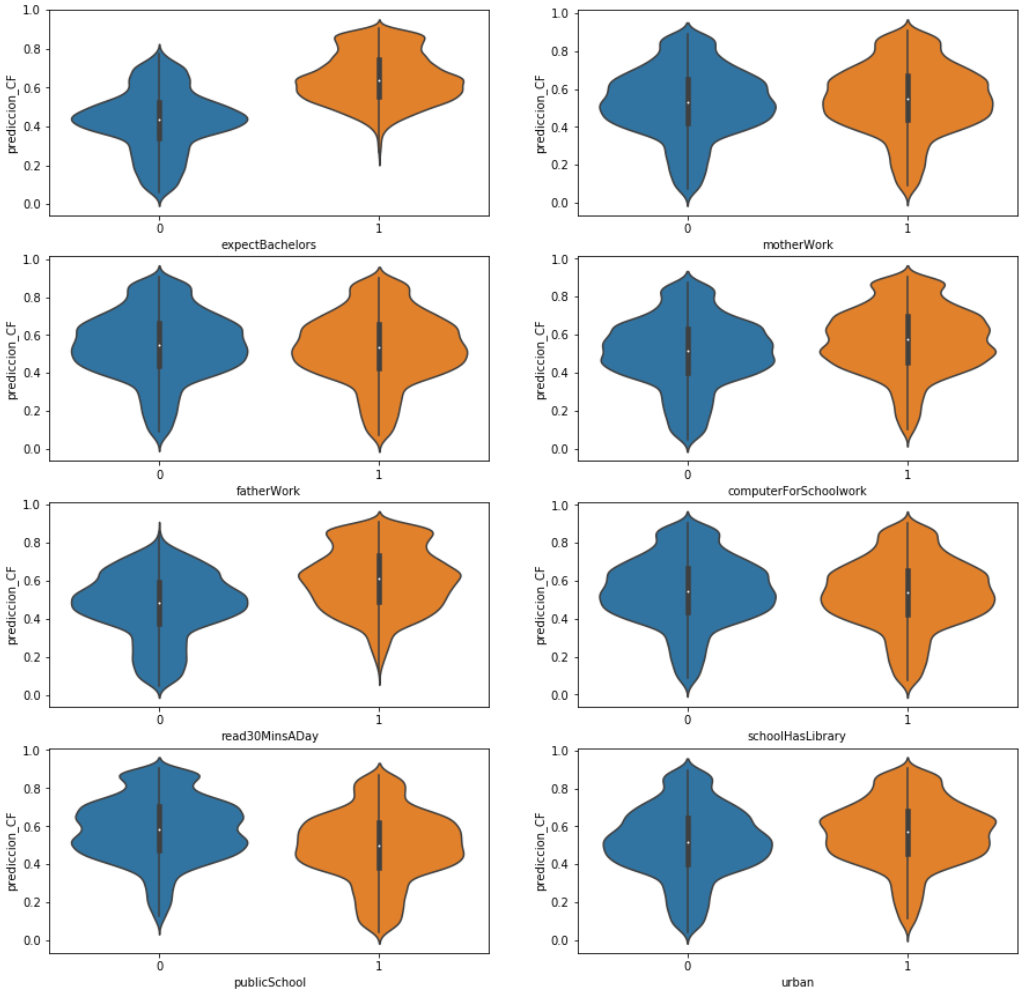
Luego de hacer las perturbaciones a las variables seleccionadas y realizar las predicciones para cada instancia contrafáctica, encontramos los siguientes resultados:

Variables continuas:



Como podemos observar, el modelo, partiendo de la instancia elegida, le asigna distintas distribuciones de probabilidad a los distintos valores de las variables (aca muestro 4 muestras de valores de cada variable). Será interesante analizar si para cualquier otro individuo las recomendaciones son parecidas.

Variables categóricas:

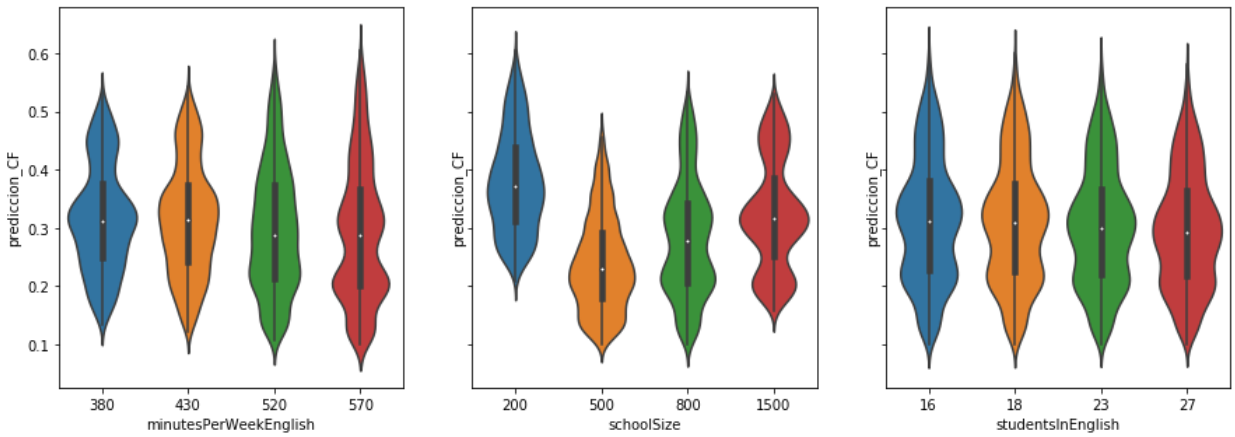


Instancia (individuo) 2:

grade	10.0
male	0.0
preschool	1.0
expectBachelors	0.0
motherHS	1.0
motherBachelors	0.0
motherWork	1.0
fatherHS	1.0
fatherBachelors	0.0
fatherWork	1.0
selfBornUS	1.0
motherBornUS	1.0
fatherBornUS	1.0
englishAtHome	1.0
computerForSchoolwork	1.0
read30MinsADay	0.0
minutesPerWeekEnglish	270.0
studentsInEnglish	18.0
schoolHasLibrary	1.0
publicSchool	1.0
urban	0.0
schoolSize	996.0
raceeth_American Indian/Alaska Native	0.0
raceeth_Asian	0.0
raceeth_Black	1.0
raceeth_Hispanic	0.0
raceeth_More than one race	0.0
raceeth_Native Hawaiian/Other Pacific Islander	0.0
raceeth_White	0.0

Resultados:

Variables continuas:



Como se ve en la figura, las distribuciones (por lo tanto las recomendaciones) cambian para el individuo 2. Notemos en schoolSize, el modelo entiende que para el individuo 1, un

tamaño de escuela más grande es mejor. Sin embargo, para el individuo 2 parece ser al revés.

Este es el potencial de las explicaciones contrafácticas: captar las influencias de las variables sobre el resultado a nivel individual.

3.4. Problema: proximidad vs diversidad

En el apartado anterior realizamos ECF de manera manual, seleccionando cambios arbitrarios en ciertas variables para simular las perturbaciones. La gran pregunta que queda pendiente es: **¿De qué forma es más conveniente perturbar las variables?**

El problema de realizar la exploración de contrafácticos de manera arbitraria es que es difícil coordinar el trade-off entre la proximidad y la diversidad. Aparte, hay muchas cosas para tener en cuenta: muchas de las variables no son factibles de ser modificadas y otras combinaciones se encuentran limitadas (no es posible agregar un título universitario y mantener constante la edad).

Para considerar todos estos factores, se presenta a continuación un conjunto de técnicas que plantea esta situación como un problema de optimización:

4. Generación de contrafácticos

TODO ESTO MEJORAR LA REDACCIÓN

4.1. Contexto y métodos anteriores

Hay otros tipos de trabajos que han querido explicar los modelos complejos usando otros métodos menos complejos:

Explicaciones a través de la importancia de features

determinar la importancia de variables mediante aproximaciones locales. LIME fitea un modelo lineal para aproximar un modelo no lineal localmente. También lo hacen con arboles de decisión. Lundberg y Lee presentan un framework que mide la importancia de las variables para cada predicción. El problema que tienen es que, como se están aproximando con modelos más simples, no son confiables del todo.

Explicaciones a traves de la visualización

similar a identificar la importancia de las variables, se puede visualizar las decisiones de un modelo. Pueden ser difíciles de interpretar.

Explicaciones a través de ejemplos

frameworks para aplicaciones basadas en ejemplos son los MMD-critic propuestos por Kim et al. Las explicaciones contrafácticas son una **perturbación** de las variables para ver su resultado usando el mismo modelo.

Contrafactico (Wachter et al.): $c=argmi\ \ nyloss(f(c),y)+|x-c|$

x = input feature, f = modelo ML, y = output del modelo.

La primera parte (yloss) empuja el contrafáctico c hacia una proyección diferente de la instancia original.
La segunda parte (|x-c|) mantiene el contrafáctico cerca de la instancia original.

Este paper extiende ese trabajo, al darle diversidad a los contrafácticos. Russell propone integer programming para modelos lineales. Acá se propone una alternativa que funciona para cualquier modelo diferenciable.

4.2. DiCE (explicaciones contrafácticas diversas)

4.2.1. Motor de generación de contrafácticos

El input de nuestro problema es un modelo ML entrenado f y una instancia x. Queremos generar un conjunto de k contrafácticos {

FORMULA

c1,c2,...,ck} tal que todos esos producen un resultado diferente que x. Todos estos ejemplos son d-dimensionales.

Aca se ASUME QUE LOS MODELOS SON DIFERENCIABLES Y ESTÁTICOS y el OUTPUT ES BINARIO.

Objetivo: generar conjunto contrafáctico FACTIBLE y accionable. Cosas a las que se puedan llegar en la realidad. Se necesitan que sean lo suficientemente diversos pero al mismo tiempo posible. Se usa la restricción de factibilidad de Wachter y otras restricciones del usuario.

4.2.2. Restricciones de diversidad y factibilidad

Diversidad mediante procesos puntuales determinantes

Se captura la diversidad mediante procesos puntuales determinantes (DPP determinantal point processes). Se usa la siguiente métrica basado en el determinante de la matriz kernel dado el contrafáctico:

FORMULA

$$dpp\_\_diversity = det(K)dpp\_\_diversity = det(K)$$

donde

K

Ki,j=11+dist(ci,cj) es la métrica de distancia entre dos ejemplos contrafácticos. Se añaden perturbaciones random a los elementos diagonales para computar el determinante.

Proximidad: ECF que están más cerca del input original son los más valiosos para un usuario. Cuantificamos la proximidad como el vector de distancia (negativo) entre el input original y cada ECF (mediante los features). Se puede usar una métrica de distancia especifica como l1 y esta se puede pesar por el usuario (puede ser hiper parámetro para cada feature). Proximidad de un conjunto de ECF es la proximidad media sobre el conjunto:

$$Proximity=-\frac{1}{k}\sum_{i=1}^{k} dist(c_i,x)$$

Proximity=−1k∑i=1kdist(ci,x)

Dispersión (sparsity): propiedad de que sea posible. Intuitivamente, un ECF va a ser más posible si cambia la menor cantidad de variables. Como es una restricción no convexa (porque siempre quieres menos), lo modificamos aparte.

Restricciones de usuarios: un ECF puede estar cerca en el espacio de features pero puede no ser posible por restricciones en la vida real. Tiene sentido permitir al usuario que delimite restricciones a la manipulación de features. Pueden especificarse de dos maneras:

- como ejemplos en cajas (usando rangos de valores para cada feature). Ejemplo: ingresos no pueden superar 200mil.
- alternativamente, el usuario puede especificar las variables que pueden ser modificadas.

TODO LO DE ARRIBA CORREGIR REDACCION Y FÓRMULAS

4.2.3. Optimización

Basado en los conceptos anteriores de diversidad y proximidad, se construye una función de costo sobre todos los contrafácticos generados:

$$C(x) = argmin_{(c_1,\ldots c_k)} \frac{1}{k} \sum_{i=1}^{k} yloss(f(c_i), y) + \frac{\lambda_1}{k} \sum_{i=1}^{k} dist(c_i, x) - \lambda_2 dppDiversity(c_1, \ldots c_k)$$

- c
- i
- ci es un ECF
- k
- k es la cantidad de ECF generados
- $f(.)$
- f(.) es el modelo ML entrenado
- $yloss(.)$
- yloss(.) es la métrica que minimiza el resultado de un ECF con el output buscado
- y
- y (usualmente 1)
- d
- d numero total de features
- x
- x es el input original
- $dppDiversity(.)$
- dppDiversity(.) es la métrica de diversidad
- λ
- 1
- λ1 y
- λ
- 2

- λ2 son hiperparametros que balancean las tres partes de la función!!!

Implementacion: optimizamos la función anterior usando **gradient descent**. Idealmente podemos obtener

f(ci)=y para cada ECF, pero a veces no es posible porque el objetivo es no-convexo. Hacemos un máximo de 5000 steps o hasta que la función de perdida converge y el contrafáctico es generado (cuando llega a la clase buscada). Se inicializan todos los ECF de forma aleatoria.

MEJORAR REDACCIÓN Y FORMULAS A LO DE ARRIBA

4.2.4. Consideraciones prácticas

Son importantes para apoyar la interacción con los usuarios de los CF.

Elección de yloss: como un contrafáctico solo necesita pasar el threshold, puede no ser necesario medir la perdida a y, sino al valor que hace pasar al threshold. O lo podemos truncar y decir que si pasa el threshold hay 0 pena (hinge loss).

Eleccion de función de distancia: para features continuos, se define la distancia como la distancia mínima l1 de los features entre el ECF y el input. Como los features pueden tener distintos rangos, dividimos por la median absolute deviation (MAD) de cada feature. Con esto captamos la importancia relativa de los cambios.. Para variables categoricas, es difícil medir la distancia. Aca se usa 1 si cambio en el valor y 0 si no.

> (se puede inferir la facilidad/posibilidad de pasar de un estado a otro según los datos de entrenamiento?)

Escala relativa de variables: como **la escala de una variable influye mucho qué tanto importa en la función objetivo**, creemos que lo ideal es permitir a los usuarios que pongan sus preferencias para las variables. Igualmente transformamos todas las features en [0,1]. Las continuas son escaladas entre 0 y 1. Para variables categóricas, convertimos cada variable con one-hot encoding y la consideramos como una variable continua.

Mejorar la dispersion: mientras la función objetivo minimiza la distancia entre el input y el ECF, el ECF ideal necesita ser diverso en la cantidad de features que cambia. Usamos una operación post-hoc cuando encuentra un ECF que va llevando de a poco los valores de nuevo al original x hasta que cambia de resultado de nuevo al no buscado.

Elección de hiperparametros: aca usamos

FORMULAS (AGREGAR)

Evaluar contrafacticos

Usualmente solo se consideran evaluaciones cualitativas. Acá se presentan cuantitativas. Últimamente, los ECF deben ayudar al usuario a entender el boundary local de decisión del clasificador de ML. Se propone una métrica que aproxima una nocion del entendimiento del usuario.

Validez, proximidad y diversidad

En primer lugar definimos métricas cuantitativas para validez, diversidad y proximidad para un conjunto contrafáctico que puede ser usado para evaluar cualquier metodo para generar contrafácticos. Asumimos que ya tenemos el conjunto C de ECF:

Validez: es la fracción de ejemplos que cumplen con el outcome buscado.

Proximidad: medimos la proximidad basado en distancias de forma separada para las variables categóricas y las continuas. Definimos la proximidad como el promedio de las distancias entre el input x y el ECF. La proximidad para un conjunto es la media de las proximidades individuales.

Dispersion: captura el número de variables que son diferentes (número de cambios entre el input original x y el ECF. También lo podemos separar para continuo y categórico.

Diversidad: de la misma forma que la proximidad, en vez de hacer la distancia al input original, se hace la distancia entre pares de ECF. Para un conjunto, se toma la media de esas distancias. Tambien se hace aparte categóricas y continuas.

Como hay tradeoffs, la evaluación de los ECF va a depender de los méritos relativos de la diversidad vs proximidad para cada aplicación particular.

Evaluación cuantitativa

Se evalua DiverseCF basado en las métricas de generación de CF de diversidad y proximidad, con los lambdas mencionados.

Explicar un modelo ML no lineal:

Validez: en todos los datasets se encuentra que DiverseCF genera casi 100% de ECF válidos para todos los valores requeridos de k (cantidad de ECF). Las otras técnicas que no tienen diversidad incluida, generan 100% solo en k=1 y después van bajando. Funciona mejor cuando hay varias variables continuas.

Diversidad: DiverseCF también genera mas ejemplos diversos que los otros métodos tanto para categóricas como continuas variables. También es el que logra mas cantidad de variables continuas modificadas (dispersión) por mas que no está incluido explícitamente como objetivo.

Proximidad: Diverse CF retorna ejemplos con menor proximidad que otros modelos sin diversidad, indicando un claro trade off entre diversidad y proximidad. Sin embargo, en muchos casos la diferencia es chica. Se le puede agregar un suplemento para mejorar la proximidad de las variables continuas post-hoc : Diverse CF-Sparse ??????? no entiendo esto. Pero es un parámetro que puede usarse para tunear si queremos más proximidad.

Evaluación cualitativa

En los tres datasets, se realizan perturbaciones de variables importantes. En el COMPÁS muestra que, con todo igual, si era blanco, el modelo predecía que no reincidir. También muestra que si era un delito menor también habría predicho que no reincidir, comparado al robo en cuestión. Estas respuestas no llevan a tomar decisiones pero proveen al usuario una imagen certera de los escenarios donde ellos habían podido salir. **En la práctica, se pueden decidir cuales son variables no modificables**.

En el dataset de income, muestra que estudiando un máster el modelo predice como que supera el umbral de ingresos, pero también muestra algunos que no son tan obvios como estar casado... (aca el problema de la causalidad). Esto se da por correlaciones generadas por variables espurias. Para resolver eso se presenta un **método de filtrado basado en relaciones causales**.

Basado en el dominio o caso particular, las personas pueden priorizar cambiar ciertas variables o tener más o menos dispersos ECF. Esto se hace **cambiando los pesos en las variables** y el learning rate para la optimización.

A TODO LO DE ARRIBA MEJORAR REDACCIÓN Y FORMULAS

5. Ejemplos con DiCE

A continuación se utilizará la librería de python DiCE para la creación de explicaciones contrafácticas. Esta librería está basada en el paper de Amit Sharma et al. que se analizó en la sección anterior sobre Explicaciones contrafácticas diversas.

Link al github: https://github.com/interpretml/DiCE/tree/master/dice_ml

Comenzamos aplicándolo al dataset de PISA USA 2009 ya usado anteriormente y, luego, presentamos dos nuevos datasets con sus respectivos procesamientos y explicaciones contrafácticas.

5.1. ECF con dataset PISA USA 2009

COMPLETAR CON LAS CONCLUSIONES

5.2. ECF con dataset PISA 2009 extenso

El dataset PISA 2009 extenso https://github.com/jbryer/pisa contiene informacion, al igual que el anterior, de las pruebas PISA del año 2009. Pero esta vez, se contienen datos de muchos otros países, no solo de Estados Unidos, así como también se cuenta con muchas otras variables.

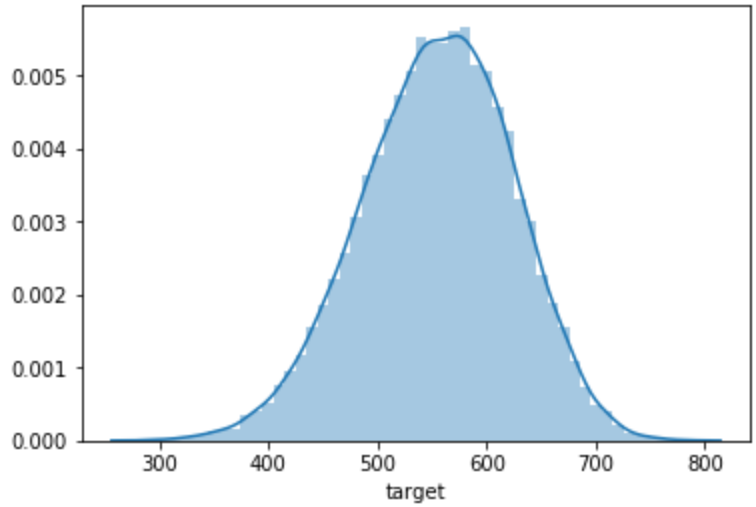
El dataset utilizado cuenta con 306 variables y mas de 475 mil observaciones. Cada fila corresponde a un estudiante que toma el examen PISA. Tambien contiene los resultados de los distintos temas (matematicas, ciencias, lectura, etc) de la evaluacion PISA.

Preprocesamiento:

1. Eliminacion de datos faltantes: el dataset tiene MUCHOS datos faltantes. El metodo utilizado para eliminarlos (ya que las redes neuronales no aceptan datasets con faltantes) es:
a) eliminar aquellas columnas que mas del 50% de sus observaciones sean NAs.
b) de lo restante eliminar las filas con datos faltantes.
cantidad que queda: Filas: 28346 , Columnas: 289

2. Corrijo el tipo de datos de las columnas. Luego del proceso, quedan 217 variables categóricas y 46 contínuas.

3. Creo un solo target: basado en el promedio de los resultados de todos los tipos de evaluaciones para cada estudiante.



4. Binarizo el outcome entre menor y mayor de 500 puntos.

5. Realizo el one-hot encoding para las variables categoricas.

6. Normalizo las variables para que queden entre 0 y 1.

Entreno la red neuronal:
PONER DATOS DE LA RED

Resultados:

Accuracy: 84.06
Area under the ROC curve : 0.874809
El punto de corte optimo es 0.7540000081062317
Confusion matrix:
[[2922 779]
[2834 10528]]
Accuracy: 0.7882552892222938
f1: 0.8535408812679881
recall: 0.7879060020954947
precision: 0.931104625453259

COMPLETAR LAS CONCLUSIONES

5.3. ECF con dataset Heart attack

Ahora se realizará el procedimiento de ECF para un dataset de ataques al corazon. Este tendrá la importancia de captar las diferencias individuales para la recomendacion de acciones para reducir la probabilidad de ataques al corazon.

https://www.kaggle.com/nareshbhat/health-care-data-set-on-heart-attack-possibility

COMPLETAR PREPROCESAMIENTO, CLASIFICACION Y RESULTADOS

6. Conclusiones y trabajos futuros

ESCRIBIR