RESUMEN

APLICACIÓN DE CIENCIA DE DATOS PARA LA CREACIÓN DE SOFTWARE PREDICTIVO DE MORBIMORTALIDAD MATERNA EN MÉXICO

por

Rúsbel Domínguez Domínguez

Asesor principal: Germán Harvey Alférez Salinas

RESUMEN DE TESIS DE MAESTRÍA

Universidad de Montemorelos

Facultad de Ingeniería y Tecnología

Título: APLICACIÓN DE CIENCIA DE DATOS PARA LA CREACIÓN DE SOFTWARE PREDICTIVO DE MORBIMORTALIDAD MATERNA EN MÉXICO

Nombre del investigador: Rúsbel Domínguez Domínguez

Asesor principal: Germán Harvey Alférez Salinas, Doctor en Informática

Fecha de terminación: Mayo de 2017

Problema

En México, la razón de mortalidad materna (RMM) es muy alta: 34.6 defunciones por cada 100 mil nacimientos estimados. Es por esto que los establecimientos y servicios de atención médica han incluido como tema prioritario la salud en el embarazo, el parto y el puerperio.

En México, existe una base de datos abierta sobre muerte materna de la Secretaría de Salud que contiene 58 variables. Dentro de estas variables se encuentra la variable “causa de clasificación internacional de enfermedades (CIE)”, que contiene un total de 248 enfermedades que están relacionadas con muertes maternas. En la actualidad, no existe un software que clasifique a las mujeres en control de embarazo

(según su riesgo de mortalidad), utilizando variables seleccionadas con ciencia de datos.

Método

El desarrollo de este proyecto está basado en la metodología que la International Business Machines (IBM) propone para la aplicación de ciencias de datos. Esta metodología se organiza en diez etapas que representan un proceso iterativo, compuesto de la siguiente manera: compresión del problema, enfoque analítico, requerimiento de los datos, recolección de los datos, comprensión de los datos, preparación de los datos, modelamiento, evaluación, despliegue y retroalimentación.

Conclusiones

Mediante la aplicación de ciencia de datos y aprendizaje automático sobre datos abiertos de la Secretaría de Salud de México, es posible clasificar a pacientes en las siguientes dos causas de muerte: eclampsia durante el trabajo de parto o padecimiento de hemorragia postparto, secundaria o tardía.

Universidad de Montemorelos

Facultad de Ingeniería y Tecnología

APLICACIÓN DE CIENCIA DE DATOS PARA LA CREACIÓN DE SOFTWARE PREDICTIVO DE MORBIMORTALIDAD MATERNA EN MÉXICO

Tesis
presentada en cumplimiento parcial de los
requisitos para el grado de
Maestría en Ciencias Computacionales

por

Rúsbel Domínguez Domínguez

Mayo de 2017

APLICACIÓN DE CIENCIA
DE DATOS PARA LA CREACIÓN DE SOFTWARE PREDICTIVO
DE MORBIMORTALIDAD MATERNA EN MÉXICO

Proyecto
presentado en cumplimiento parcial de
los requisitos para el grado de
Maestría en Ciencias Computacionales

por

Rúsbel Domínguez Domínguez

APROBADO POR LA COMISIÓN:

Germán Harvey Alférez

Dr. Germán Harvey Alférez Salinas
Asesor principal

Dra. Ana Lidia Santos Chong
Asesor externo

Dra. Verenice Zarahí González Mejía
Miembro

Dra. Raquel B. de Korniejezuk
Directora de Estudios de Posgrado

M.C.T.I. Saulo Hernández Osoria
Miembro

Mayo 2017
Fecha de aprobación

DEDICATORIA

A Dios, por su gran amor por mí y porque siempre está a mi lado.

A mis padres, por sus oraciones, su esfuerzo, su dedicación, su apoyo y motivación constantes al llevar a cabo cada uno de mis proyectos.

A mis hermanos Jharley y Genny, mis dos grandes y mejores ejemplos.

A mis amigos, quienes supieron brindarme su atención y aprecio, apoyándome en cualquier momento.

En forma no menos especial, a todos mis mentores, quienes formaron parte de mi proyecto educativo en la Universidad de Montemorelos.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

RECONOCIMIENTOS

A mi Dios y Padre, por guiarme, instruirme, animarme y acompañarme durante toda mi vida.

Al doctor Germán Harvey Alférez Salinas, mi asesor principal, por su apoyo, tiempo, dirección y paciencia en la elaboración de esta tesis. Porque me respaldó desde el inicio hasta el final de este trabajo.

A la doctora Verenice González, por darse el tiempo necesario para orientarme en todo lo relacionado con el área de la salud, pero también por el cariño que ha demostrado hacia mí en toda esta aventura de la realización de mi tesis.

Al maestro Saulo Hernández Osoria, quien sin su ayuda, definitivamente no habría sido posible elaborar este trabajo de investigación en el desarrollo del software, por su incansable ayuda, por su tiempo y dedicación.

A la maestra Rosa G. Grajeda Ordóñez, por dedicar el tiempo necesario para la revisión de este documento.

A la maestra Martha A. Olivas Dyk, por su apoyo para que esta investigación tuviera el formato y estilo correctos. Por su tiempo, por su comida cuando fue necesario y su apoyo en general.

CAPÍTULO I

INTRODUCCIÓN

Planteamiento del problema

La muerte materna se define como la muerte de una mujer mientras está embarazada o dentro de los siguientes 42 días siguientes a la terminación de un embarazo (Auditoria Superior de la Federación, 2013). Esta definición incluye muertes maternas que tienen una causa directa o indirecta en el embarazo.

Desde el inicio de este milenio, la Organización Mundial de Salud (OMS) se ha enfocado en disminuir y eliminar esta razón de muerte (Organización Mundial de la Salud, 2015). La OMS (2015) y la Secretaría de Salud (2016, párr. 1) menciona que se

> entiende esta mejora como todo proceso o herramienta dirigida a reducir la brecha a nivel sistémico y organizacional bajo los principios básicos de la calidad, que incluyen la atención centrada en la persona, la mejora continua de los procesos y la seguridad del paciente como prioridades para el fortalecimiento de los sistemas de salud.

No obstante, la mortalidad materna es inquietantemente alta. En México, la razón de mortalidad materna (RMM) es de 34.6 defunciones por cada 100 mil nacimientos estimados (Auditoria Superior de la Federación, 2013).

Esta razón de muerte no solamente muestra el estado actual de la salud femenina dentro de cada país, sino que también indica cuán eficaces son los programas de salud de cada país. Se entiende que muchas muertes pudieron haber sido evitadas pero,

por alguna razón, pasan desapercibidas hasta que las complicaciones en el embarazo causan otra muerte. Las muertes maternas no solamente están relacionadas con causas biológicas; también pueden estar asociadas a variables sociales (por ejemplo, edad de la madre, ocupación, estado civil) y clínico-administrativa (por ejemplo, asistencia médica o acceso al servicio de salud) que pueden hacer la diferencia entre la vida o la muerte de una madre.

Declaración del problema

En México, la dirección general de información en salud (DGCES) opera una base de datos abierta sobre muerte materna que ha sido creada con datos recopilados en cada estado de la República Mexicana por parte de la Secretaría de Salud. Esta base de datos está organizada en 58 variables o características. Dentro de estas variables, se encuentra la variable “causa de clasificación internacional de enfermedades (CIE)” que contiene un total de 248 enfermedades que están relacionadas con muertes maternas.

En la actualidad, en México no existe un software que clasifique a las mujeres en control de embarazo (según su riesgo de mortalidad), utilizando variables seleccionadas con ciencia de datos.

La contribución de este trabajo es encontrar las variables que pueden ser utilizadas para realizar una clasificación, mediante la aplicación de ciencia de datos (en una forma retrospectiva), a una base de datos abierta de la Secretaría de Salud; Esto con el fin de construir un software que contenga esas variables para clasificar a embarazadas con riesgo de mortalidad.

La generación del modelo predictivo será mediante el entrenamiento realizado con aprendizaje automático. El proceso que se seguirá para la administración de los

datos está basado en la metodología de ciencia de datos de IBM (Rollins, 2015), en la que se señalan las etapas para la aplicación de la metodología. En esta investigación, se cubren todos los pasos de esta metodología para seleccionar las variables sobre las cuales se realiza la predicción de enfermedades de alto riesgo para las mujeres embrazadas mediante la creación de un modelo predictivo basado en aprendizaje automático.

Las contribuciones específicas se enlistan a continuación:

1. Mediante ciencia de datos y aprendizaje automático se analizaron las variables de la base de datos de muerte materna en México que permitieron realizar la predicción de enfermedades de alto riesgo para las mujeres embrazadas.

2. Mediante aprendizaje automático se crearon modelos predictivos para las enfermedades encontradas como causantes de muerte materna. Estos modelos se evaluaron con el fin de crear un software que utilice el modelo predictivo más acertado.

3. Se desarrolló un software con el que se puede predecir el riesgo que tiene una paciente de presentar las siguientes dos causas de muerte: eclampsia durante el trabajo de parto o padecimiento de hemorragia postparto (secundaria o tardía).

Justificación

Los indicadores de salud en un país son el reflejo del desarrollo del mismo. Uno de los indicadores más importantes es la salud materno-infantil, ya que esta influye mucho en la distribución de variables sociodemográfico-económicas o de la pirámide poblacional. Este indicador repercute en la distribución laboral por género, de demanda de educación y de salud del país.

En México, la RMM es muy alta (34.6 defunciones por cada 100 mil nacimientos estimados). Es por esto que la DGCES, a través del programa de acción específico

2013 - 2018 titulado “Estrategia Nacional para la Consolidación de la Calidad en los Establecimientos y Servicios de Atención Médica”, ha incluido como tema prioritario la salud en el embarazo, el parto y el puerperio.

En México existe una base de datos sobre muerte materna de la Secretaría de Salud que contiene 58 variables. Dentro de estas variables se encuentra la variable “causa de CIE”, que contiene un total de 248 enfermedades que están relacionadas con muertes maternas. En la actualidad no existe un software que utilice esta base de datos para facilitar el análisis de los factores de riesgo que pueden causar la muerte materna.

Objetivos

Objetivo general

El objetivo general de esta investigación es construir un software para la clasificación de riesgo de muerte materna en México mediante la aplicación de ciencia de datos y aprendizaje automático.

Objetivos específicos

1. Reducir el número de variables de riesgo de muerte materna mediante el uso de ciencia de datos y aprendizaje automático.

2. Crear modelos predictivos para la clasificación de enfermedades que pueden llevar a la muerte materna mediante aprendizaje automático.

3. Evaluar los modelos de clasificación generados.

4. Crear un software que utilice el modelo predictivo con los mejores resultados para clasificar a mujeres embarazadas de acuerdo con riesgos de mortalidad.

Hipótesis

Un software que utiliza modelos de predicción de enfermedades de muerte materna generados mediante la aplicación de ciencia de datos y de aprendizaje automático sobre datos abiertos públicos de muerte materna en México puede ser utilizado para clasificar a mujeres embarazadas de acuerdo con factores de riesgo que puedan causarles la muerte.

Limitaciones

No tener acceso a la base de datos actualizada de muerte materna en México por parte de la Dirección General de Información en Salud. Este estudio se realizó con datos que comprenden del año 2002 al año 2013.

Delimitaciones

Con el fin de alcanzar una solución oportuna y práctica al problema planteado, la investigación se delimitó de la siguiente manera:

1. Se desarrolló un prototipo de software para el ingreso de datos de pacientes embarazadas que retorna la clasificación de la paciente de acuerdo con factores de riesgo que puedan causarles la muerte.

2. Se utilizó el lenguaje de programación Python y la librería Scikit Learn para la construcción de los modelos predictivos.

3. Los modelos se evaluaron mediante la utilización de mecanismos de validación cruzada.

Definición de términos

Es común encontrar en la literatura palabras o términos desconocidos. A

continuación, se han seleccionado términos no comunes presentes en este trabajo, con la finalidad de contextualizar al lector.

API: Es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software. La sigla API viene del inglés application programming interface. En español, es interfaz de programación de aplicaciones.

Aprendizaje automático: El aprendizaje automático (machine learning) es un campo de estudio de la inteligencia artificial que consiste en desarrollar técnicas que permiten a las computadoras aprender, utilizando información estructurada o no estructurada (Camargo Vega, Camargo Ortega y Joyanes Aguilar, 2015).

Backend: Es todo el procesamiento y programas que no ven.

BSD: Berkeley software distribution.

Característica o feature: Rasgo distintivo de una clase.

CIE: Clasificación internacional de enfermedades.

Ciencia de datos: La ciencia de datos se enfoca en la obtención de conocimiento de los datos. Esta ciencia se basa en las técnicas y teorías de la estadística, las matemáticas, la programación y la inteligencia artificial (Dhar, 2013).

Clase: Refiere atributos de la(s) variable(s) dependiente(s).

DGCES: Dirección general de calidad y educación en salud.

DGIS: Dirección general de información en salud

FIT: Facultad de ingeniería y tecnología de la Universidad de Montemorelos.

Fronted: Es la visualización de la interfaz gráfica de usuario.

IBM: International Business Machines Corp.

ILDA: Iniciativa latinoamericana de datos abiertos.

INEGI: Instituto Nacional de Estadística y Geografía.

Instancia: Son los registros de la base de datos que se describen mediante características.

Inteligencia artificial: Es un conjunto de técnicas que, mediante el empleo de la informática, permite la realización automática de operaciones hasta ahora exclusivas de la inteligencia humana (Real Academia Española, 2014).

UML: Unified modeling language (lenguaje unificado de modelado).

OCDE: Organización para la Cooperación y el Desarrollo Económicos.

OMS: Organización Mundial de la Salud

Python: Es un lenguaje de programación que permite trabajar de forma rápida, integrando los sistemas con mayor eficiencia.

Scikit Learn: Es un módulo de Python que integra una amplia gama de algoritmos de aprendizaje automático.

Software: Es el conjunto de instrucciones legibles por máquina que dirige una computadora para realizar operaciones específicas.

CAPÍTULO II

MARCO TÉORICO Y TRABAJO RELACIONADO

Marco teórico

En esta sección se presenta el marco teórico en donde se explican los siguientes conceptos: mortalidad materno-infantil en el mundo, mortalidad materno-infantil en las Américas, mortalidad materno-infantil en México, aprendizaje automático, aprendizaje supervisado y no supervisado, algoritmos de aprendizaje supervisado, aprendizaje no supervisado mediante el análisis de componentes principales, ciencia de datos, metodología de ciencia de datos, datos abiertos, Python, Scikit Learn, uso de ciencia de datos en salud pública, trabajo relacionado en salud pública en la Facultad de ingeniería y tecnología (FIT) de la Universidad de Montemorelos (UM).

Mortalidad materno-infantil en el mundo

Las complicaciones en el embarazo, únicamente en el año 2010, fueron responsables de 287,000 muertes en el mundo (Organización Mundial de la Salud, 2015). El hecho de que existan tantas muertes maternas en una época de grandes avances médicos habla de lo ineficientes que son los programas de salud (Potter, 2014). Se ha encontrado que la mayoría de estas muertes se presentan en países en donde la mayor parte de la población vive con un salario bajo o mediano (Moon, 2010).

Una investigación realizada en la Habana, Cuba muestra que los países industrializados registran solamente 14 casos de muerte materna por cada 100,000 bebés

nacidos vivos. Por el contrario, los países en vía de desarrollo pueden llegar a tener hasta 290 muertes maternas por cada 100,000 bebés nacidos vivos (Potter, 2014; Quiroz Huerta, Tepetla, Cortés Salazar, Rojo Contreras y Morales Andrade, 2015). Esta desigualdad en casos de muertes en países industrializados y países en vía de desarrollo habla de la importancia en la identificación de las características detonantes con la morbimortalidad materna en una población.

Lo que hace la diferencia en la estabilidad de las embarazadas es que, entre más rápido se pueda diagnosticar un problema de salud, menor es la probabilidad de que una paciente pueda desarrollar una complicación que pueda terminar en una defunción (Córdova Villalobos et al., 2008; Segura, 2014).

Mortalidad materno-infantil en las Américas

Las mujeres en las Américas que viven en condiciones de pobreza, en sitios alejados, con menor nivel de educación, indígenas y de otros grupos étnicos o raciales y, además, en situación de violencia de género, tienen una sobrerrepresentación en la razón de muerte materna. Se reconoce que un 95% de la mortalidad materna podría ser evitada con el conocimiento y la tecnología ya disponibles en la región (Organización Panamericana de Salud, 2014).

Desde el año 2014, la Organización Panamericana de la Salud (OPS) y la Organización Mundial de la Salud (OMS) se han propuesto promover intervenciones exitosas de promoción, prevención, diagnóstico y tratamiento oportuno y adecuado para reducir las muertes maternas a través del continuo cuidado del individuo, la familia y la comunidad que acuden a los servicios de salud (Organización Panamericana de Salud, 2015).

Mortalidad materno-infantil en México

México presenta un problema multifactorial en salud pública relacionado tanto con la cobertura como con la calidad de los servicios de salud que impacta en la morbimortalidad (Layton, Campillo, Carrete, Ablanedo Terrazas y Sánchez Rodríguez, 2010; Organización para la Cooperación y el Desarrollo Económicos, 2016). En México, las principales causas de muerte materna son las siguientes: hemorragia obstétrica, preclamsia, eclampsia, complicaciones del aborto y sepsis puerperal que, en conjunto, representan el 68% del total de las defunciones maternas (Auditoria Superior de la Federación, 2013).

En los últimos años, en México se ha puesto énfasis en tomar en cuenta variables sociales tales como la economía de los pacientes y su estatus social. Se ha encontrado que la falta de educación formal está asociada a un incremento de seis veces, en el riesgo de padecer una complicación durante el embarazo (LaFleur y Vélez, 2014).

Contar con herramientas que faciliten la gestión de salud en las pacientes, con base en sus riesgos particulares, podría ayudar a la disminución de la mortalidad materna, como lo afirma la Organización Mundial de la Salud (2015).

Aprendizaje automático

Aprendizaje automático (machine learning en inglés) es una disciplina científica del ámbito de la inteligencia artificial para crear sistemas que aprenden automáticamente con base en conocimiento previo (Camargo Vega et al., 2015). Aprender en este contexto quiere decir identificar patrones complejos en datos. Una máquina que aprende, es capaz de predecir comportamientos futuros (Michalski, Carbonell y Mitchell, 2013).

Una de las aplicaciones más recientes del uso de aprendizaje automático es el

estudio de enfermedades raras, las cuales presentan dificultad al momento de identificarlas (Ramos, González-Alcaide y Gutiérrez, 2016). El tener una mayor disponibilidad de datos de pacientes podría mejorar el entrenamiento de algoritmos de aprendizaje automático y permitiría hacerle frente al problema de identificación de enfermedades mediante el uso de la inteligencia artificial (Garg, Dong, Shah y Jonnalagadda, 2016).

Aprendizaje supervisado y no supervisado

El aprendizaje automático está dividido en dos tipos: el aprendizaje supervisado y el aprendizaje no supervisado.

El aprendizaje supervisado consiste en la creación de una función capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos llamados datos de entrenamiento (González, 2015). Para ello, la función tiene que generalizar a partir de los datos presentados a las situaciones no vistas previamente; es decir, se pueden descubrir las relaciones entre variables independientes (también llamadas atributos o características) y para realizar predicciones que puedan ser útiles en la toma de decisiones.

Se requiere explicar algunos conceptos para que se logre una correcta comprensión del aprendizaje supervisado: (a) el término “clase” refiere a la variable dependiente, (b) “característica” (o feature en inglés) refiere a una variable independiente, un rasgo distintivo de una clase y (c) “instancias” son los registros de un conjunto de datos. Cada instancia se describe en términos de sus características y su clase.

El aprendizaje no supervisado es un método donde un modelo es ajustado a las observaciones. Este se distingue del aprendizaje supervisado por el hecho de que no

hay un conocimiento a priori. Así, el aprendizaje no supervisado típicamente trata los objetos de entrada como un conjunto de variables aleatorias, siendo construido un modelo de densidad para el conjunto de datos (Hinton y Sejnowski, 1999). Antes de aplicar los métodos de aprendizaje automático se deben identificar las siguientes características en los datos.

Algoritmos de aprendizaje supervisado

En esta sección se describen los siguientes algoritmos de aprendizaje supervisado utilizados en esta investigación: k nearest neighbors (KNN), support vector machine (SVM), naïve bayes y regresión logística.

K nearest neighbors (KNN)

KNN es un algoritmo que clasifica cada nueva muestra basándose en la etiqueta de los vecinos más cercanos (K), determinada a partir de distancias euclidianas a la nueva muestra. El algoritmo consiste en la creación de *"n" clusters* (racimos) o también denominadas clases, definidos por el usuario (Garrido, 2015). El algoritmo realiza la clasificación de un nuevo dato en función de la probabilidad de que este pertenezca a una de las clases especificadas. Este es un método ampliamente reconocido (Maillo, Ramírez, Triguero y Herrera, 2016).

Como se observa en la Figura 1, se tienen dos clases representadas con dos símbolos diferentes (cuadrados y triángulos). Ante una nueva entrada (representada por el círculo), el algoritmo tomará la decisión de clasificar a este dentro de una de las clases en función de la probabilidad de que pertenezca a dicha clase.

A continuación, se muestra un ejemplo de una clasificación con KNN (ver Figura 1).

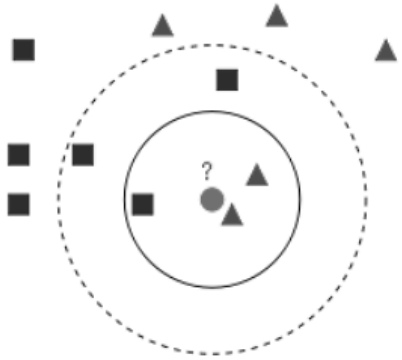

Figura 1. Ejemplo de una clasificación con KNN (adaptación de Antti Ajanki, 2007).

Naïve bayes (NB)

Este es un clasificador probabilístico fundamentado en el Teorema de Bayes y algunas hipótesis simplificadoras adicionales (Ng y Jordan, 2002).

La Figura 2 muestra un ejemplo de este algoritmo. Los clientes tienen una buena separación entre ellos (se puede ver una clara distinción entre buenos y malos clientes). Puede suponerse que los triángulos están más cerca del nuevo cliente (cuadrado) que los círculos y, por lo tanto, es más probable que la nueva clase a la cual pertenezca este cliente sea la voluntad de ser un buen cliente.

Para medir esta probabilidad, se dibuja un círculo alrededor de la nueva observación (el cuadrado), que abarca un número (elegido a priori) de puntos, independientemente de sus etiquetas de clase (Hernández Orallo, Ramírez Quintana y Ferri Ramírez, 2004). A continuación, se cuenta el número de puntos pertenecientes a cada clase en el círculo. A partir del resultado, se calcula la probabilidad. La probabilidad de que el cliente sea bueno responde a la fórmula que figura como Ecuación 1 ($\alpha = \frac{3}{20}$), y la probabilidad que el cliente sea malo responde a la fórmula que figura

como Ecuación 2 ($\alpha = \frac{1}{10}$).

Mediante el uso de la regla de bayes, se calcula una nueva probabilidad llamada "la probabilidad posterior", que es una combinación de la probabilidad a priori y de la probabilidad. La probabilidad posterior de que un cliente sea bueno es igual a $\left\{\frac{20}{30} * \frac{3}{20} = 0.1\right\}$.

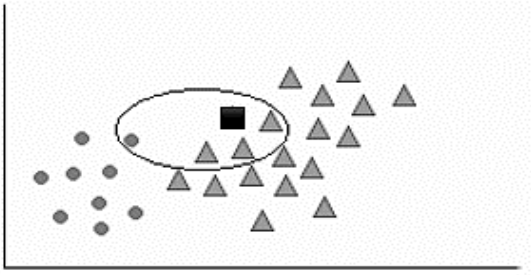

Figura 2. Ejemplo de clasificación con naïve bayes (Sankar, 2015).

La probabilidad posterior de que un cliente sea malo o bueno es igual a $\left\{\frac{10}{30} * \frac{1}{10} = 0.03\right\}$.

Dado que la probabilidad de que el nuevo cliente sea bueno es mayor, el algoritmo de naïve bayes clasifica este punto de datos como un nuevo cliente.

Support vector machine (SVM)

El SVM funciona mediante la búsqueda de puntos que se pueden utilizar como "vectores de apoyo" para definir los límites de clasificación entre las clases.

Este algoritmo utiliza núcleos lineales y gaussianos. El kernel gaussiano se basa en una cartografía de características infinitas (Herrera Zurita y Duran Cals, 2016).

En la Figura 3 se crea un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permite una buena clasificación (Garrido, 2015).

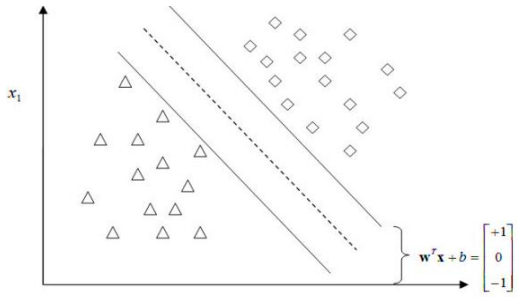

Figura 3. Ejemplo de una clasificación con SVM (Ho, 2009).

Regresión logística

La regresión logística modela la probabilidad de tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica. Esta es una variable que puede obtener un número limitado de categorías en función de las variables independientes (Herrera Zurita y Duran Cals, 2016). Puede ser usado para modelar la probabilidad de que ocurra un evento, así como función de otros factores.

El análisis de regresión logística se enmarca en el conjunto de modelos lineales generalizados (Agresti, 2002). De tal forma, la regresión logística es adecuada cuando la variable de respuesta Y es politómica (permite diversas categorías de

respuesta, tales como mejora mucho, empeora, se mantiene, etc.), pero es especialmente útil cuando solo hay dos posibles respuestas (cuando la variable de respuesta es dicotómica), que es el caso más común (De la Fuente Fernández, 2011).

La regresión logística es una de las técnicas estadístico-inferenciales más empleadas en la producción científica contemporánea (ver Figura 4).

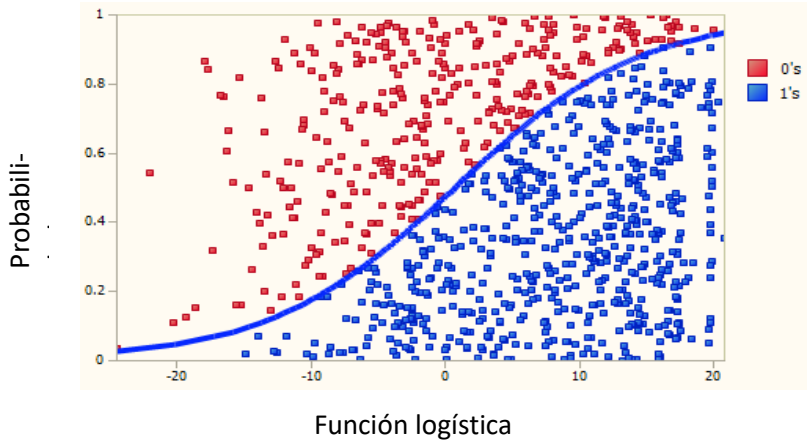

Figura 4. Ejemplo de una clasificación con regresión logística (fuente propia).

La regresión logística puede utilizarse para tratar de correlacionar la probabilidad de una variable cualitativa binaria (que puede tomar valores reales "0" y "1") con una variable escalar x. Se espera que la regresión logística aproxime la probabilidad de obtener "0" (no ocurre cierto suceso) o "1" (ocurre el suceso) con el valor de la variable explicativa x (De la Fuente Fernández, 2011).

Aprendizaje no supervisado mediante análisis de componentes principales (PCA)

El análisis de componentes principales (PCA) es un filtro que permite la

transformación de los datos y que utiliza un método de búsqueda, con el cual se procura realizar una reducción de dimensiones eligiendo las características que den un porcentaje de varianza en los datos y que filtra el ruido de los atributos eliminando las características. Se emplea sobre todo en análisis de ciencia de datos y para construir modelos predictivos (Shlens, 2014).

Indiscutiblemente, la razón principal por la que se transforman los datos en un análisis de componente principal es para comprimir los datos mediante la eliminación de la redundancia. Un ejemplo concreto de redundancia en datos es evidente en un ráster multibanda que está constituido por elevación, pendiente y orientación. Debido a que la pendiente y la orientación generalmente se derivan de la elevación, la mayoría de las varianzas dentro del área de estudio se pueden explicar solamente mediante la elevación (Irimia-Dieguéz, Blanco-Oliver y Oliver-Alfonso, 2016).

Ciencia de datos

La ciencia de datos es una disciplina emergente multidisciplinaria que comprende desde el desarrollo de software, hasta la inteligencia artificial, el manejo de datos y la estadística (Dhar, 2013). Los proyectos con ciencia de datos se basan en identificar correlaciones, relaciones causales, clasificar y predecir eventos, identificar patrones y anomalías e inferir probabilidades, Estos proyectos se enfocan en la obtención de conocimiento de los datos y están estrechamente relacionados con datos masivos (big data en inglés) (Schoenherr y Speier-Pero, 2015). La ciencia de datos cubre el proceso de la recopilación de los datos, su análisis y los resultados de los experimentos.

Con la creciente capacidad de recolectar, almacenar y analizar una diversidad cada vez mayor de datos, la ciencia de datos se está incrementando rápidamente.

Promueve el uso de algoritmos que pueden generar resultados útiles. Por desgracia, en el campo de la ciencia de datos no se conoce el "mejor" proceso para realizar un proyecto científico (Saltz, Shamshurin y Crowston, 2017).

Metodología de ciencia de datos

IBM (International business machines) propone una metodología para la aplicación de ciencia de datos a nivel industrial y científico (Rollins, 2015). Esta metodología se organiza en diez etapas que representan un proceso interactivo (ver Figura 5), las cuales se presentan a continuación.

1. Compresión del proyecto. Es la etapa donde se definen tanto los problemas y los objetivos del proyecto como también los requisitos de la solución desde un punto de vista comercial. Por eso, esta etapa se considera como la más difícil.

2. Enfoque analítico. En esta etapa, el científico de datos define el enfoque analítico que utilizará para resolver el problema. Esto implica saber expresar el problema en el contexto de técnicas estadísticas y de aprendizaje automático para que pueda identificar los mecanismos adecuados para lograr un resultado exitoso.

3. Requerimiento de datos. La elección de la aproximación analítica se determina a partir de los requisitos de los datos para poder conocer el método analítico que se utilizará. Las características de los datos son particulares en diversos formatos y representaciones; pueden ser estructurados, semi-estructurados y no estructurados.

4. Recolección de datos. En esta etapa, el científico de datos identifica y reúne datos estructurados, semi-estructurados o no estructurados, relevantes en el dominio del problema.

5. Comprensión de los datos. En esta etapa, la estadística descriptiva y las técnicas

de visualización pueden ayudar a entender al científico de datos el contenido de los datos, evaluar su calidad y descubrir una idea inicial de estos.

6. Preparación de los datos. Esta etapa comprende actividades enfocadas en la preparación del conjunto de datos que serán usados en la etapa de modelado. Esto incluye la limpieza de los datos, la combinación de los datos de múltiples fuentes y la transformación de los datos.

7. Modelamiento. En esta etapa, se generan modelos predictivos mediante algoritmos de aprendizaje automático.

8. Evaluación. En esta etapa, el científico de datos evalúa la calidad de los modelos generados.

9. Despliegue. Después de haber elegido un modelo satisfactorio que es aprobado por los patrocinadores del proyecto, el modelo se implementa en el ambiente de producción o de prueba para su utilización.

10. Retroalimentación. Mediante la recopilación de los resultados del modelo implementado, la organización recibe información sobre el rendimiento del modelo y se observa la forma en que este afecta su entorno. El análisis de esta información le muestra al científico de datos si se debe refinar el modelo. Esto aumenta su precisión y, por tanto, su utilidad.

Datos abiertos

Los datos abiertos pueden ser utilizados, reutilizados y redistribuidos libremente por cualquier persona. Se encuentran sujetos, a lo sumo, al requerimiento de atribución y de compartirse de la misma forma en que aparecen inicialmente (Valdivia, Navarrete y Aracena, 2014).

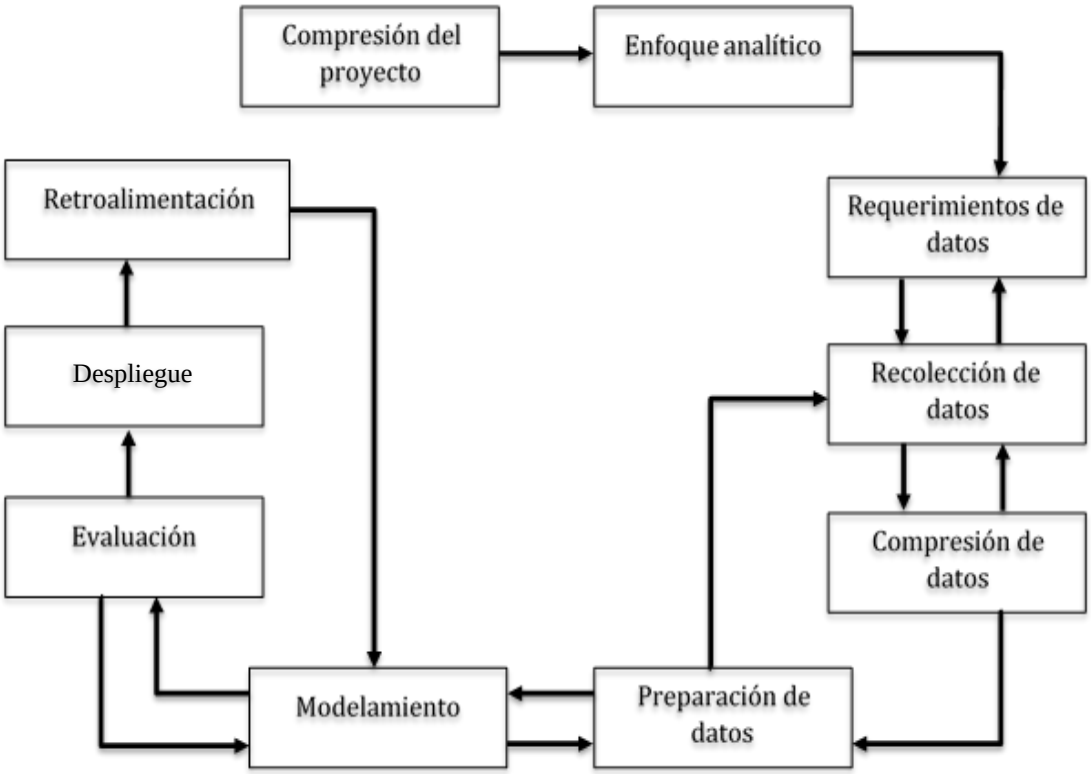

Figura 5. Metodología fundamental para la ciencia de datos de IBM (Rollins, 2015).

En el año 2015 fue fundada “La Carta Internacional de Datos Abiertos” en colaboración de gobiernos y de expertos que convinieron en seis principios de cómo los gobiernos deben publicar información. La aspiración compartida fue que los datos deben ser abiertos por defecto, oportuna e interoperable.

Más de 70 gobiernos y organizaciones se han unido al movimiento (Open data chárter, 2013).

La Revista u-GOB, que se especializa en temas de innovación y tecnología en el gobierno, reconoció a México por su colaboración en el comité directivo de la carta internacional de datos abiertos. Desde este comité, México ha colaborado con

organizaciones como la iniciativa latinoamericana de datos abiertos (ILDA) para impulsar la adopción de principios de apertura de la información en gobiernos de toda la región (Coordinación de Estrategia Digital Nacional, 2017).

Se conoce también de la existencia de repositorios de datos abiertos en investigaciones. Su utilización contribuye no solo a la transparencia, pues permiten comprobar si los métodos y resultados de una investigación se han realizado de acuerdo con la cultura científica de cada área, sino que además permiten avanzar a la ciencia puesto que pueden suponer ahorro de tiempo y dinero al reutilizar recursos producidos por otros (Hernández Pérez y García Moreno, 2013).

Python

Python es un lenguaje de programación interpretativo, interactivo y orientado a objetos. Proporciona estructuras de datos de alto nivel, tales como listas y arreglos asociativos (denominados diccionarios), sintaxis dinámica, vinculación dinámica, módulos, clases, excepciones, gestión de memoria automática, entre otras (Vinent, Milián, Estrada y Bonet, 2016)

El lenguaje Python tiene una sintaxis notablemente simple y elegante; sin embargo, es un lenguaje de programación potente y de propósito general. Fue diseñado en 1990 por Guido Van Rossum (Ramos Guadarrama, Hernández Areu y Bruzón Hernández, 2016). Python es gratuito y puede ser utilizado con fines comerciales, así como también puede ser ejecutado en prácticamente cualquier computadora moderna.

Python incluye una diversa biblioteca de extensiones estándar (algunas escritas en Python, otras en C o en C++) para realizar operaciones que van desde manipulaciones de cadenas y expresiones regulares de tipo Perl, hasta gráficos de interfaz de

usuario que incluyen utilidades relacionadas con la Web, como servicios del sistema operativo, depuración, herramientas de creación de perfiles, entre otras (Ramírez, 2010).

Asimismo, Python permite crear nuevos módulos para extender el lenguaje con código nuevo o heredado. De hecho, hay un número considerable de módulos escritos en Python que han sido desarrollados y distribuidos por miembros de la comunidad (Van Rossum, 2009). Estos módulos de extensión son denominados "paquetes" que pueden ser descargados por separado. Algunos ejemplos de ellos son los siguientes: PIL, la biblioteca de imágenes de Python; CORBA Compliant Object Request Brokers (ORB) escrito en Python; GadFly, un gestor de base de datos SQL escrito en Python; y Gendoc, una herramienta de documentación automatizada (Fan et al., 2016; Mhenni, Nguyen y Choley, 2016).

El mejor recurso para Python junto con los libros que están disponibles es su propio sitio web (http://www.python.org). Este sitio proporciona acceso a códigos, documentación, paquetes, artículos, listas de correo, entre otros.

Scikit Learn

Scikit Learn es un módulo de Python que integra una amplia gama de algoritmos de aprendizaje automático con tecnología de última generación para resolver problemas de escala media de aprendizaje supervisado y no supervisado (Scikit-Learn, 2016).

Scikit Learn es fácil de utilizar, es rápido, tiene muy buena documentación y es consistente. Cuenta con dependencias mínimas y se distribuye bajo la licencia de Berkeley Software Distribution (BSD), para favorecer su utilización, tanto en el ámbito

académico como en el comercial. El código fuente, el código binario y la documentación de Scikit Learn pueden descargarse de su sitio web (http://scikit-learn.org/stable/).

Trabajo relacionado

Uso de ciencia de datos en salud pública

Día a día los profesionales de la salud en todo el mundo se esfuerzan en encontrar la manera de mejorar la calidad de vida en la población. Sin embargo, el mundo está cambiando a un ritmo acelerado con la aparición o reaparición de enfermedades infecciosas tales como el zika, el ébola, la chikungunya y la fiebre amarilla (Ramos et al., 2016). Actualmente, durante el diagnóstico y tratamiento, el personal médico no solamente se basa en pruebas clínicas; también incursiona en la medicina basada en evidencias científicas (de Dios, Sempere y Aleixandre-Benavent, 2007). Es aquí donde radica la importancia de ejecutar un análisis de los datos clínicos que permitan efectuar predicciones con las cuales se pueda realizar un aporte en la salud.

La utilización de la ciencia de datos está generando resultados positivos debido a su capacidad de automatizar trabajo cognitivo de forma rápida y precisa. Este hecho se evidencia, por ejemplo, de que al utilizar datos recopilados de twitter, existe una amplia capacidad de predecir epidemias, como se observó poco tiempo atrás en el caso de la influenza (Guerrero, 2016; Lazer, Kennedy, King y Vespignani, 2014).

Un ejemplo de la aplicación de ciencia de datos en el área clínica lo muestra un estudio realizado por científicos de la Universidad de Stanford, quienes usaron aprendizaje automático para clasificar cuatro tipos de cáncer en riñón y de pulmón basados en sus perfiles de metilación. Ellos encontraron que mediante regresión logística y SVM

se puede lograr una precisión cercana al 95% al realizar la clasificación (Jain, Zhou y Men, 2013).

En el campo de la oftalmología, investigadores de Stanford han usado aprendizaje automático para clasificar la retinopatía diabética. Específicamente, ellos exploraron diversas técnicas de aprendizaje automático para la identificación y la clasificación de gravedad de retinopatía diabética en imágenes de retina (Honnungar, Mehra y Joseph, 2016).

La Encuesta Nacional de Altas de Hospitales realizada por el gobierno de los Estados Unidos reveló que del año 2000 al año 2010 se realizaron 51,4 millones de procedimientos clínicos (Hall, Levant y DeFrances, 2013). De igual forma, indicó que en el año 2011 se recibieron 125,7 millones de personas en el área de consulta externa y 136,3 millones de atenciones en el área de urgencias. Estos datos son una muestra de la cantidad de datos que se obtienen de los centros hospitalarios. La ciencia de datos puede ser utilizada para el análisis sistemático de estos datos con la finalidad de generar un aporte a la sociedad en el área de salud.

La importancia de la ciencia de datos se evidencia en proyectos tales como “Datasphere”. En este proyecto, empresas farmacéuticas se han unido para compartir, integrar y analizar conjuntos de datos de experimentos relacionados con el cáncer, con el propósito de generar resultados y acelerar la creación de una cura (Hede, 2013).

Por otro lado, un organismo sin fines de lucro impulsado por la Universidad de Chicago, bajo el nombre de “data science for social good”, ha realizado estudios con ciencia de datos, utilizando un conjunto de datos proporcionados por el gobierno mexicano relacionados con las características que están presentes en muertes maternas.

Los datos fueron recaudados por el INEGI. Dicho estudio ayuda a conocer dónde hay problemas potenciales y cuáles son estos. Así como también permitirá a México mejorar las intervenciones y el diseño de programas de salud con la finalidad de salvar a las madres mexicanas (Eng, 2016).

Asimismo, el Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional en México realizó una comparación de técnicas de selección de las características para clasificar lesiones de mama en ultrasonografía mediante el uso de aprendizaje automático. En este estudio, se comparó la capacidad de discriminar, del conjunto de características completo, cualquier número de características o atributos y los subconjuntos, determinados por técnicas de análisis de componentes principales (PCA) e información mutua (MI), con cualquier cantidad de atributos mediante la prueba estadística de Kruskal-Wallis (Meza, 2014).

Uso de ciencia de datos en la FIT de la UM

En relación con el uso de ciencia de datos en el área de la salud, en el año 2014, investigadores de la FIT desarrollaron un software para la clasificación de nevos sospechosos de cáncer, usando reconocimiento de imágenes y redes neuronales artificiales. A través de esta aplicación, médicos en zonas rurales y remotas podrán llevar a cabo detecciones tempranas de nevos sospechosos de cáncer de una manera rápida, sencilla, automatizada, no invasiva y de bajo costo (Marín, Alférez, Córdova y González, 2015).

Por otra parte, a pesar de los asombrosos adelantos relacionados con las comodidades y el bienestar de la vida, y aun con la higiene y el tratamiento de las enfermedades, resulta alarmante el aumento de los padecimientos que impactan a nivel

epidemiológico. Por ejemplo, de acuerdo con la Organización Mundial de la Salud, 422 millones de adultos sufren diabetes.

En México, la diabetes es un problema de salud pública altamente relevante. En 2015 se contabilizaron 11 millones de casos de diabetes, convirtiéndola en la principal causa de muerte en mujeres y la segunda en hombres desde el año 2000. Es por esto que, en otra investigación, Alférez et al. (2016a,b), contribuyeron a corroborar la relación entre diabetes y caries dental encontrada en la literatura, mediante la aplicación de técnicas de ciencia de datos en historias clínicas de la Clínica Dental de la Universidad de Montemorelos. Este trabajo se conectó con la tendencia actual de las diferentes instituciones relacionadas con la ciencia. Esta tendencia consiste en establecer estrategias y servicios efectivos de promoción de la salud, los cuales pueden cristalizarse en una cultura de corresponsabilidad y autocuidado de la población en lo referente a la diabetes y su impacto en la salud oral.

CAPÍTULO III

METODOLOGÍA

Introducción

En este capítulo se presenta la descripción de los pasos seguidos en esta investigación con base en la metodología de ciencia de datos de IBM (Rollins, 2015).

Comprensión del proyecto

La Secretaría de Salud de México cuenta con una base de datos abierta con datos de cada estado, del 2002 al 2013, sobre muerte materna. Esta base de datos está organizada en 58 variables. Dentro de estas variables, se encuentra la variable “causa de clasificación internacional de enfermedades (CIE)” que contiene un total de 248 enfermedades que están relacionadas con muertes maternas. Los valores de la variable CIE fueron considerados como clases para el desarrollo de este proyecto. En otras palabras, la variable CIE es dependiente.

Pensado en el aporte de este proyecto a la sociedad, puede notarse que, en la actualidad, no existe un software en México que clasifique a las mujeres en control de embarazo (según su riesgo de mortalidad), utilizando variables seleccionadas con ciencia de datos.

Enfoque analítico

Con la finalidad de analizar los datos de muerte materna, se decidió utilizar

algoritmos de aprendizaje supervisado, debido a que existe la necesidad de realizar la clasificación de las mujeres en control de embarazo según su riesgo de mortalidad con base en los datos abiertos de la Secretaría de Salud de México

Requerimientos de datos

Cada uno de los 14,308 registros o instancias en el conjunto de datos obtenido de la Secretaría de Salud está descrito mediante un conjunto de características o variables independientes (ver Apéndice A). Estas características son descriptivas (tipo de dato String) o numéricas (tipo de dato Integer). Sin embargo, en esta investigación únicamente se utilizaron las características numéricas, pues los algoritmos de aprendizaje automático utilizados únicamente trabajan con números

El archivo con los datos puede verse en la Figura 6 en su formato original (CSV). Este archivo puede ser descargado en el siguiente sitio web: (https://drive.google.com/open?id=0BzO8zQQuN9yjSnJ3QWFYQWZ0WVk).

Recolección de los datos

En esta investigación, se utilizó el conjunto de datos abiertos provisto por la Secretaría de Salud de México (Dirección General de Información en Salud, 2016). Este conjunto de datos cuenta con 14,308 registros de muertes maternas desde el año 2002 hasta el año 2013 y con un total de 58 variables descritas en el Apéndice A. Este conjunto de datos se puede descargar del sitio Web de la Secretaría de Salud de México (http://www.dgis.salud.gob.mx/descargas/datosabiertos/mortalidad_materna_ 2002-2013.zip).

Figura 6. Ejemplo de la base de datos de muerte materna de la Secretaria de Salud de México en formato CSV.

Comprensión de los datos

En este paso, se utilizó microsoft excel para facilitar la comprensión del conjunto de datos obtenidos del DGIS. Con el fin de comprender mejor los datos, se comparó el conjunto de datos obtenidos del DGIS con los datos públicos del INEGI. Los resultados de esta comparación fueron muy cercanos (ver Tabla 1). No obstante, el INEGI no presenta las instancias particulares de defunciones maternas. Por esto, se decidió utilizar el conjunto de datos del DGIS.

Preparación de los datos

Existen técnicas de limpieza y preparación de los datos que son importantes para el éxito de proyectos basados en datos, tal como describen Han, Pei y Kamber (2011). Este es uno de los pasos más demorados en el proceso de ciencia de datos.

Tabla 1

Comparación de las muertes maternas registradas por año de dos bancos de datos: INEGI y DGIS.

Año	INEGI	DGIS
2002	Sin datos	1348
2003	1752	1350
2004	1539	1278
2005	1631	1287
2006	1584	1204
2007	1427	1157
2008	1493	1167
2009	1594	1281
2010	1338	1078
2011	1336	1067
2012	1057	1073
2013	Sin datos	1019

Las técnicas que se utilizaron en esta investigación para la preparación de los datos se enuncian a continuación.

Errores en datos y validación de datos vacíos o nulos

Los datos se exportaron a una hoja de cálculo de microsoft excel, con el fin de aplicar filtros para ordenar los datos de mayor a menor con respecto a cada una de las características en el conjunto de datos. Esto sirvió para encontrar “valores atípicos”, los cuales son sospechosos de no pertenecer al conjunto de datos de donde proceden o ser producto de algún suceso sumamente extraño. En este paso se redujo el número de registros de 14,308 a 14,279.

Validación de los tipos de datos

En este paso se eliminaron las características descriptivas de tipos de datos String, debido a que los algoritmos de aprendizaje automático utilizados en esta investigación únicamente trabajan con valores numéricos. En el caso de la clase “causa de CIE”, los valores de cada enfermedad se especificaron en números de forma manual (ver Apéndice B).

Validación de los valores

Los valores numéricos deben ser coherentes. En la Figura 7 se puede observar el filtrado realizado en una hoja de cálculo de microsoft excel. En este caso, se revisó la columna G de la hoja de excel que muestra la edad cumplida de cada paciente. En esta figura se puede observar que hay registros con valores atípicos con el valor 998. En total, se eliminaron 29 registros con este problema, dejando un total de 14,279 en el conjunto de datos.

	A	B	C	D	E	F	G	H	I
1	Año de na	Mes de na	Descripció	Día de nac	Genero	Clave de l	Edad cum	Estado co	Descripció
2	0	0	NO ESPECIFI	0	2	A	998	0	SE IGNORA
3	0	0	NO ESPECIFI	0	2	A	998	5	CASADO
4	0	0	NO ESPECIFI	0	2	A	998	5	CASADO
5	0	0	NO ESPECIFI	0	2	A	998	0	SE IGNORA
6	0	0	NO ESPECIFI	0	2	A	998	5	CASADO
7	0	0	NO ESPECIFI	0	2	A	998	0	SE IGNORA
8	0	0	NO ESPECIFI	0	2	A	998	5	CASADO
9	0	0	NO ESPECIFI	0	2	A	998	0	SE IGNORA
10	0	0	NO ESPECIFI	0	2	A	998	5	CASADO
11	0	0	NO ESPECIFI	0	2	A	998	5	CASADO
12	0	0	NO ESPECIFI	0	2	A	998	0	SE IGNORA
13	0	0	NO ESPECIFI	0	2	A	998	4	UNION LIBRE

Figura 7. Ejemplo del filtrado para valores no coherentes.

Modelamiento

En esta etapa se generaron modelos predictivos mediante aprendizaje automático que pudieron servir para predecir enfermedades de alto riesgo en mujeres embarazadas. Al inicio del proyecto, se contempló la utilización del software de licencia libre “Weka” por ser un programa popular para el análisis y procesamiento de datos (Riza, 2015).

No obstante, al realizar experimentos de clasificación, se encontró que en la versión 3.6 de Weka de microsoft windows 10 pro, no se encuentran habilitados los algoritmos de SVM y regresión logística. Para resolver estos problemas, se descargó la librería “Libsvm-3.21” con la finalidad de poder acceder correctamente a SVM desde Weka, pero no se logró tener resultados positivos.

Fue así que se decidió incursionar en el lenguaje de programación Python, en el cual se pudieron desarrollar los modelos de clasificación con SVM, KNN, naïve bayes, y regresión logística. Estos algoritmos se eligieron por ser muy utilizados para el análisis de datos mediante aprendizaje supervisado (Han et al., 2011). En este caso, la CIE fue considerada como la clase de cada instancia en el conjunto de datos. Asimismo, en este estudio se contempló la utilización del análisis de componentes principales (PCA, por sus siglas en inglés) con el fin de tratar de reducir el número de características a utilizar en los experimentos.

Evaluación

En esta etapa, se evaluaron los experimentos desarrollados con el grupo de algoritmos de aprendizaje supervisado elegidos en el paso anterior: SVM, KNN, naïve bayes y regresión logística. Con la finalidad de encontrar el mejor modelo predictivo al

ejecutar estos algoritmos, se calculó el accuracy, la precision, el recall y el F1 de cada modelo generado. Para realizar esto, se dividio el conjunto de datos en dos: el 70% de estos datos sirvió para realizar el entrenamiento, mientras que el 30% fue utilizado para la evaluación del modelo generado.

A continuación, se describen los conceptos de accuracy, precision, recall y f1-score.

Accuracy

La exactitud (accuracy en inglés) es una medida adicional de precisión que se define como el promedio de la fracción de aciertos del clasificador frente a la unión de etiquetas reales y predichas. Se muestra la exactitud con un valor de 1 o, lo contrario, con un valor de 0. En otras palabras, es la comparación entre los valores verdaderos y los valores entregados en un valor numérico (Accuracy_score, 2017).

Precision

La precisión (Precision en inglés) de un clasificador que se define como el cociente entre las etiquetas correctamente asignadas y todas las etiquetas asignadas por el clasificador. Una puntuación alta en precisión indica que el clasificador está realizando clasificaciones correctamente (Garrido, 2015). Según Precision-Recall (2016), la precisión (*P*) es determinada por el número de verdaderos positivos (*Tp*) entre el número de verdaderos positivos más el número de falsos positivos (*Fp*). La fórmula queda de la siguiente forma: $\left\{P = \frac{T_p}{T_p + F_p}\right\}$.

Recall

El alcance (recall en inglés) de un clasificador es el cociente entre las etiquetas correctamente clasificadas y las realmente positivas. En otras palabras, el recall (*R*) es determinado por el número de verdaderos positivos (*Tp*) entre el número de verdaderos positivos más el número de falsos negativos (*Fn*) (precision-Recall, 2016).

La fórmula queda de la siguiente manera: $\left\{R = \frac{T_p}{T_p+F_n}\right\}$.

F1-score

El F1-score se define como la media armónica entre la precision y el recall. Se obtiene al realizar la siguiente operación: {2 * (Precision * Recall) / (Precision + Recall), (f1_score, 2017)}, quedando la fórmula como se muestra a continuación: $\left\{F1 = 2\frac{P \times R}{P+R}\right\}$.

Despliegue

En este paso se construyó un software para poder predecir el riesgo que tiene una paciente en estado de embarazo de presentar una enfermedad que pueda llevarla a la muerte. Con este fin, se eligió el modelo predictivo con los mejores resultados de la evaluación del paso anterior para utilizarlo en la clasificación de la paciente. El backend de este programa computacional se desarrolló en Python. El frontend se escribió en PHP y HTML.

Python se eligió por ser una herramienta potente y comúnmente utilizada en el área de ciencia de datos y datos masivos (Davenport y Patil, 2012). En este proyecto se utilizó la versión 2.7 de Python y la versión 4.1.1 de Anaconda (plataforma impulsado por Python cuya versión es de código abierto y con una capacidad de alto rendimiento en Python y R). El ambiente de desarrollo se ejecutó sobre microsoft windows

de 64 bits, sobre el cual se instalaron la versión 0.17.1 de Scikit-Learn y la versión 0.10.1 de Blaze. Estas herramientas son muy populares en el desarrollo de proyectos de inteligencia artificial y ciencia de datos con tecnología de última generación para la resolución de problemas, mediante aprendizaje supervisado y no supervisado.

Modelos UML

En esta sección se presentan dos diagramas UML para una mejor comprensión del funcionamiento del software desarrollado en este trabajo de investigación.

Diagrama de casos de uso

En el diagrama de casos de uso de la Figura 8 se muestran tres actores: médico, paciente y científico de datos. El médico es quien realiza una consulta a la paciente para realizar un diagnóstico; el paciente comparte la información solicitada con el médico para recibir un diagnóstico y el científico de datos es quien realiza el análisis de los datos y crea un modelo de clasificación que se utiliza durante la ejecución del software.

A continuación, se describen brevemente cada uno de los casos de uso.

1. Ejecutar el clasificador: el científico de datos crea un modelo de clasificación mediante algoritmos de aprendizaje supervisado.

2. Evaluar el clasificador: el científico de datos evalúa los modelos de clasificación. El mejor modelo es el que se ejecuta en el backend del software.

3. Introducir los datos de la paciente al sistema: el médico llena un formulario con los siguientes datos de la paciente: mes de nacimiento, día de nacimiento, edad cumplida, estado conyugal, entidad de residencia, tamaño de localidad, ocupación habitual,

escolaridad, derechohabiencia, entidad, municipio, localidad y asistencia médica

4. Realizar inferencia: el algoritmo realiza la inferencia de la posible enfermedad de la paciente con base en el modelo entrenado.

5. Recibir el diagnóstico: el médico recibe el diagnóstico en pantalla. A continuación, puede referir a la paciente al médico especialista para tratar oportunamente el caso.

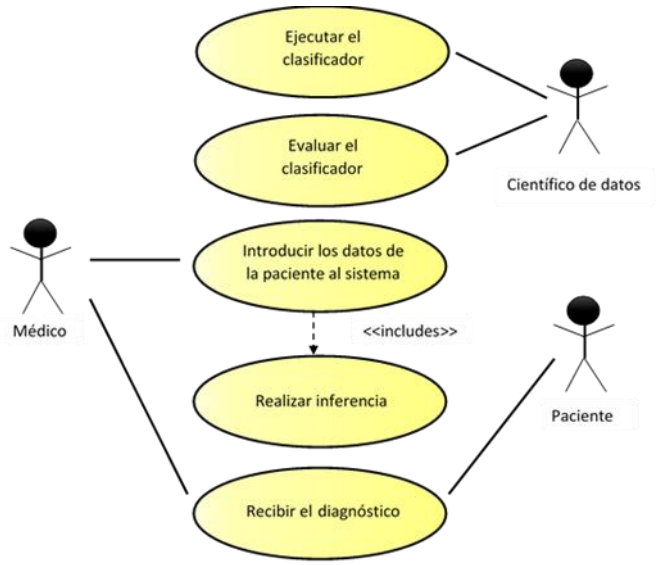

Figura 8. Diagrama de casos de uso.

Diagrama de secuencia

En la Figura 9 se muestra el diagrama de secuencia UML de los módulos presentes en el producto de software. El actor médico abre un formulario en el cual introducirá los datos de la paciente (form.html). Una vez completado el cuestionario, el formulario envía los datos a costom.js, el cual asigna valores numéricos a cada valor

introducido para poder ser analizado por el modelo de clasificación.

Proccess.php inicia Python.exe para correr el módulo EjecutarModelo.py. EjecutaModelo.py realiza una clasificación con los nuevos valores contenidos en un arreglo y entrega el nombre de la clase en la cual se clasificaron los nuevos datos. El resultado se envía a form.html, el cual muestra al médico la clasificación de la paciente mediante un mensaje de alerta.

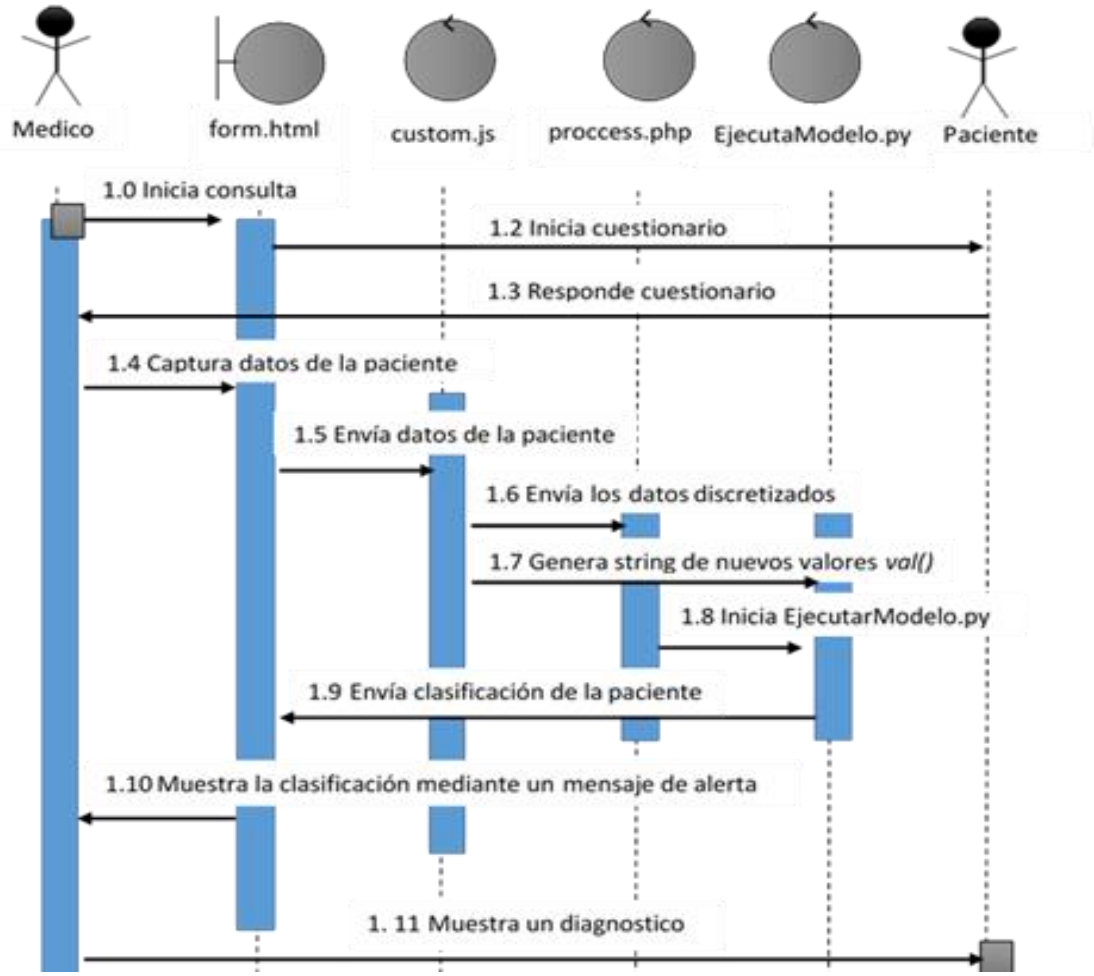

Figura 9. Diagrama de secuencia.

En la Figura 10 se muestra el prototipo de la interfaz del software desarrollado para la captura de información de pacientes embarazadas. Esta interfaz muestra el cuestionario, que está compuesto por trece preguntas. La mayoría de estas preguntas tienen inmediatamente debajo de cada una de ellas la opción de seleccionar la información requerida, con excepción del día de nacimiento y la edad cumplida, ya que estos dos campos son alimentados manualmente y tienen un rango límite.

El mes de nacimiento comprende del día 1 al día 31. La edad cumplida comprende de 12 a 55 años. En este último caso, se limitó a estas edades, ya que son los valores con los cuales fue entrenado el algoritmo.

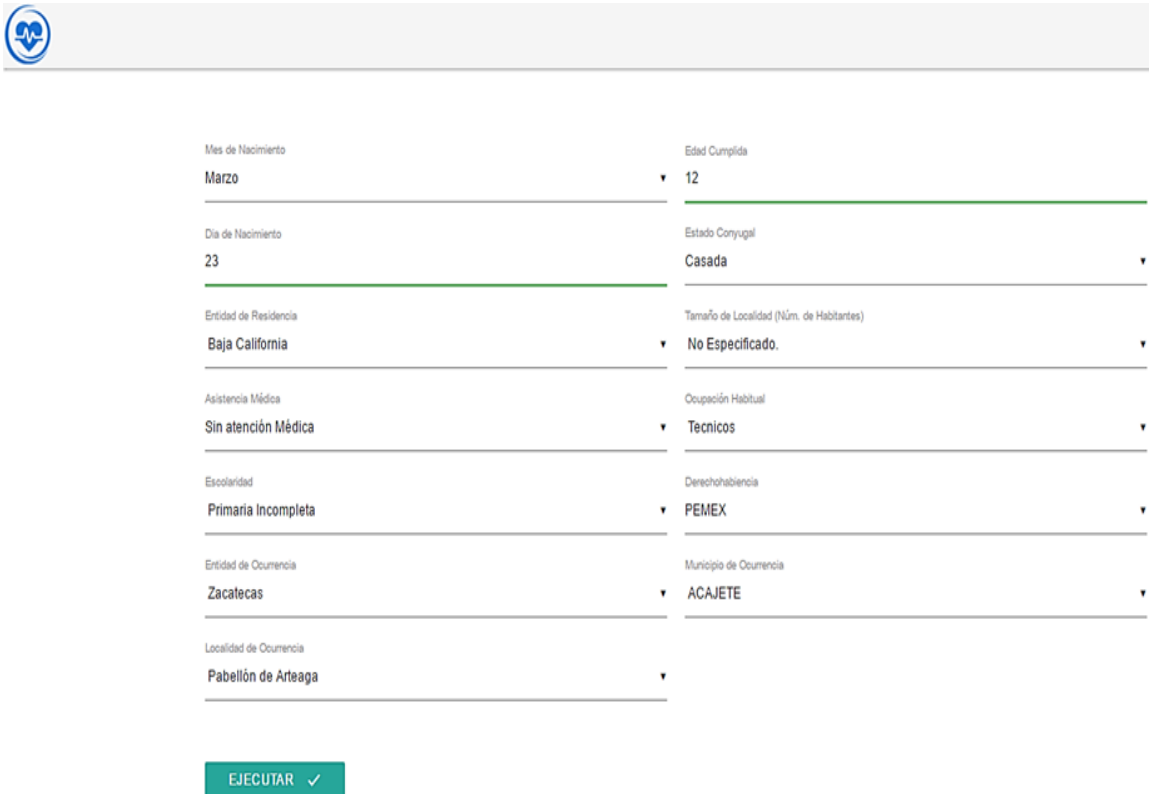

Figura 10. Interfaz del producto de software para clasificación de mujeres embarazadas.

Los datos se estructuran en el formato JSON para poder ser mostrados por form.html. El Apéndice C muestra los códigos completos de cada uno de los módulos. Asimismo, estos módulos están disponible en la siguiente dirección web: (https:// drive.google.com/drive/folders/0BzO8zQQuN9yjZGlOSkp3bU50NGM?usp= sharing).

El código *EjecutarMdelo.py* en Python toma los datos ingresados por parte del personal médico y realiza la conversión de texto (String) a números (Integer), lo que se conoce como discretización de los datos, para poder ser analizados por los algoritmos, así como se muestra en el módulo de Custum.js en el Apéndice C. Para poder realizarse la inferencia, se ejecuta el módulo entrenado *EjecutarModelo.py*.

Finalmente, el sistema muestra el resultado de la clasificación al médico (ver Figura 11).

El mensaje retornado contiene el nombre de la enfermedad que la paciente puede padecer durante el embarazo y en los siguientes 42 días después del alumbramiento. Este mensaje se muestra a través de un mensaje de alerta.

Figura 11. Mensaje de alerta que muestra el resultado de la clasificación.

Retroalimentación

En la etapa de retroalimentación, el investigador y el director de esta investigación analizaron los resultados de los posibles modelos predictivos generados mediante los algoritmos clasificadores. Con este conocimiento, se realizaron modificaciones en

los experimentos se ejecutaron nuevos experimentos, y se llegó al modelo predictivo más adecuado para ser ejecutado en el backend de la herramienta.

CAPÍTULO IV

PRESENTACIÓN Y ANÁLISIS DE RESULTADOS

Esta sección presenta los resultados de la aplicación de un conjunto de algoritmos de aprendizaje supervisado a los datos abiertos de mortalidad materna provistos por la Secretaría de Salud de México. Los experimentos realizados sirvieron para encontrar el algoritmo que pudiera generar el mejor modelo de clasificación de enfermedades que pueden causar la muerte materna.

Ambiente de entrenamiento y pruebas

El entrenamiento y la evaluación fueron realizados en Python versión 2.7, corriendo en windows 10 pro (64 bits), en una Laptop HP con un procesador AMD Athlon™ II P320 Dual-Core a 2.10 GHz y con una memoria RAM de cuatro MB. El Apéndice D presenta los códigos escritos para realizar el entrenamiento y las evaluaciones de los modelos generados. Asimismo, en el Apéndice C se muestran los códigos para realizar las inferencias y la interfaz de usuario. Todos los códigos fuente están disponibles en la siguiente dirección web: (https://drive.google.com/drive/folders/0BzO8zQQuN9yjZGlOSkp3bU50NGM?usp= sharing).

Resultados

En el desarrollo de este proyecto, se realizaron experimentos con cuatro algoritmos de aprendizaje supervisado, los cuales son los siguientes: SVM, KNN, regresión

logística y naïve bayes. Este grupo de algoritmos fueron escogidos por ser los algoritmos populares para realizar clasificaciones (Bahgat, Rady y Gad, 2016; Herrera Zurita y Duran Cals, 2016; Shukla, Gupta y Prasad, 2016; Wang et al., 2016).

Además, se aplicó PCA con la finalidad de reducir el número de características que pudieran ser utilizadas para el entrenamiento. El argumento consiste en que un menor número de variables a ingresar por parte del personal médico causaría una reducción en el tiempo previo en la generación del diagnóstico. Es así que, aparte de generar modelos con la aplicación de algoritmos individuales, también se ejecutaron en conjunto con PCA: SVM + PCA, KNN + PCA, regresión logística + PCA y naïve bayes + PCA.

En todos los experimentos se utilizó el 70% del conjunto de datos para realizar el entrenamiento y el 30% de los mismos para realizar las evaluaciones del modelo entrenado.

En la Figura 12 se muestra la secuencia de los experimentos realizados. Se explica brevemente a continuación, teniendo en cuenta que el objetivo de estos experimentos es encontrar qué clase se logra clasificar mejor al utilizar alguno de los ocho algoritmos:

1. En el experimento 1, se ejecutaron los ocho algoritmos, contemplando las 14,279 instancias y con 248 clases.

2. El experimento 2 consistió en no tomar en cuenta las clases con menos de cien instancias, ya que estas no permiten un correcto entrenamiento de los algoritmos.

3. En el experimento 3, se utilizó el resultado obtenido del experimento 2, con un total de 10,802 instancias que pertenecen a 28 clases. Cada clase tiene cien o más instancias en ella (Beleites, Neugebauer, Bocklitz, Krafft y Popp, 2013).

4. En el experimento 4, fue necesario reducir el número de clases y de instancias, debido a que aún no se lograba una correcta clasificación con 28 clases. En este paso, quedaron cinco clases mejor que cubren un total de 4,429 instancias.

5. En el experimento 5, se obtuvieron tres clases, las cuales contienen un número de 1,600 instancias con las cuales se ejecutaron los ocho algoritmos con la finalidad de encontrar las clases mejor clasificadas y saber mediante qué algoritmo se logra esa clasificación.

6. El experimento 6 está dividido en tres experimentos (6A, 6B, 6C). En el experimento 6A, se probó con las clases 52 y 54 que contienen 1,157 instancias. En el experimento 6B, se probó con las clases 52 y 163 que contienen 1,018 instancias. En el experimento 6C, se probó con las clases 54 y 163 que contienen 1,025 instancias. Estas tres clases se eligieron por ser las que mejor se clasificaban.

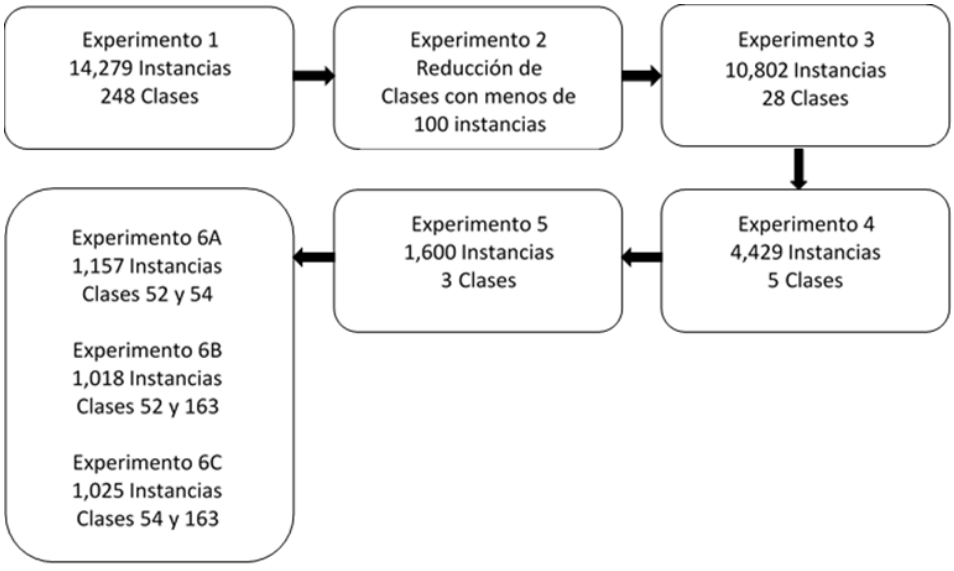

Figura 12. Diagrama de la secuencia de los experimentos realizados.

Experimento 1

En este experimento, se utilizó el conjunto de datos de muerte materna con 248 clases y 14,279 instancias. La Tabla 2 muestra la comparación de los resultados obtenidos con cada uno de los algoritmos ejecutados. Los resultados muestran el promedio de los resultados para todas las clases con respecto a precision, recall, f1-score, score y accuracy.

Tabla 2

Resultados del análisis de ocho diferentes algoritmos para el análisis de los datos de muerte materna

Algoritmo Clasificador	Accuracy	Precision	Recall	F1-Score
KNN	0.7182	0.06	0.06	0.06
PCA + KNN	0.7163	0.06	0.06	0.06
Naïve bayes	0.0153	0.05	0.01	0.01
PCA + Naïve bayes	0.0130	0.04	0.00	0.00
Regresión Log.	0.1170	0.05	0.11	0.06
PCA + Regresión Log.	0.1250	0.05	0.11	0.05
SVM	0.8948	0.04	0.08	0.02
PCA + SVM	0.6619	0.04	0.08	0.02

En este caso, SVM es el algoritmo que muestra un mayor accuracy, con un valor cercano a 1. La exactitud o accuracy del modelo es el número de predicciones correctas del total de predicciones realizadas. No obstante, el accuracy por sí sola, por lo general, no da suficiente información para tomar una decisión acerca de la robustez de las predicciones (Brownlee, 2014). Por lo tanto, es necesario contar con otras medidas tales como precision y recall. Una baja precision puede indicar un alto número de falsos positivos. Por otra parte, un bajo recall indica un alto número de falsos negativos. En este caso, los

valores de precision y recall en todos los casos son bajos.

Durante este experimento, se encontró que, de las 248 clases iniciales, 51 de estas correspondían a una sola instancia. Por este motivo, los algoritmos no pudieron entrenarse adecuadamente. De hecho, para tener una aceptable clasificación es necesario contar con por lo menos cien instancias, así como señalan Beleites et al. (2013).

Experimento 2

Este experimento consistió en realizar la eliminación de clases que no cumplían con los requerimientos para una buena clasificación, según indican Beleites et al. (2013). Los resultados de esta depuración se muestran en la Tabla 3. De un total de 248 clases descritas en el Apéndice D, únicamente 28 clases cuentan con cien o más instancias. Esto permitió reducir el número de instancias de 14,279 a 10,802.

Experimento 3

En este experimento se utilizaron los resultados del experimento 2 con instancias pertenecientes a 28 clases (ver Tabla 4). La Tabla 5 muestra las cinco clases con una mejor clasificación. La columna denominada “clase” indica el número de la clase que obtuvo la mejor clasificación en términos de accuracy en la ejecución de los ocho algoritmos que contenían un total de 28 clases. La columna “número de algoritmos” señala el número de veces que cada clase fue bien clasificada dentro del grupo de ocho algoritmos. La columna “mejor accuracy” indica el algoritmo con el cual se obtuvo el valor más alto de accuracy al realizar los experimentos. Las cuatro clases obtenidas son las siguientes: 52 (eclampsia durante el trabajo de parto), 54 (eclampsia en período no especificado), 163 (hemorragia postparto secundaria o tardía)

y 229 (trastornos mentales y del comportamiento leves, asociados con el puerperio, no clasificados en otra parte). El resultado de los experimentos realizados, se puede consultar en la siguiente dirección web: (https://drive.google.com/open?id=0BzO8zQQuN9yjazZMdkU5NS05M2M).

Tabla 3

Clases que contienen cien o más instancias y el número de instancias en cada clase

Clase	Núm. de Instancias
4	221
7	141
50	101
52	1219
53	134
54	1264
56	308
57	367
78	162
104	146
117	393
120	204
154	167
155	109
163	1055
164	1006
165	135
184	322
195	345
212	415
213	167
214	112
224	155
225	595
226	446
227	305
229	781
238	241

Tabla 4

Resultados obtenidos al experimentar con instancias en 28 clases

Algoritmo Clasificador	Accuracy	Precision	Recall	F1-Score
KNN	0.9994	0.08	0.08	0.08
PCA + KNN	0.9996	0.09	0.09	0.09
Naïve bayes	0.1524	0.12	0.14	0.11
PCA + Naïve bayes	0.1524	0.11	0.11	0.09
Regresión Log.	0.1558	0.14	0.15	0.09
PCA + Regresión Log.	0.1623	0.10	0.17	0.10
SVM	0.9409	0.05	0.12	0.05
PCA + SVM	0.9409	0.11	0.12	0.06

Tabla 5

Selección de clases por tendencia a una mejor clasificación

Clase	Núm. de algoritmos	Mejor accuracy
52	7	KNN
54	7	KNN
163	6	KNN
229	3	KNN
155	1	SVM

Experimento 4

De acuerdo con los resultados del experimento 3, en el presente experimento se aplicó una vez más el grupo de algoritmos sobre el conjunto de datos con instancias correspondientes a las clases 52, 54, 155, 163 y 229 (ver Tabla 5).

Los resultados de este experimento se muestran en la Tabla 6. En este caso, los clasificadores KNN y KNN+PCA presentan las accuracies más altas. Sin embargo,

los resultados para precision y recall son bajos. Esto demuestra que la simple evaluación del accuracy es insuficiente en este tipo de evaluaciones.

Experimento 5

En este experimento se tomaron las tres mejores clases clasificadas por cada algoritmo, aplicando el mismo método utilizado en el experimento 3. Las clases que tuvieron la tendencia a ser mejor clasificadas son las clases 52 (eclampsia durante el trabajo de parto), 54 (eclampsia, en período no especificado) y 163 (hemorragia post-parto secundaria o tardía). Con estas tres clases se aplicó nuevamente el grupo de algoritmos. En este caso, ningún resultado fue óptimo en términos de precision y recall (ver Tabla 7).

Tabla 6

Resultados de la aplicación del grupo de algoritmos al conjunto que contiene cinco clases resultantes del experimento 3

Algoritmo Clasificador	Accuracy	Precision	Recall	F1-Score
KNN	1.00	0.34	0.34	0.34
PCA + KNN	1.00	0.33	0.33	0.33
Naïve bayes	0.3769	0.38	0.38	0.37
PCA + Naïve bayes	0.3769	0.40	0.37	0.35
Regresión Log.	0.3827	0.35	0.37	0.34
PCA + Regresión Log.	0.3884	0.34	0.36	0.35
SVM	0.9887	0.28	0.28	0.19
PCA + SVM	0.9887	0.31	0.30	0.20

Tabla 7

Resultados de la aplicación del grupo de algoritmos al conjunto de datos que contiene las clases 52, 54 y 163

Algoritmo Clasificador	Accuracy	Precision	Recall	F1-Score
KNN	1.00	0.40	0.40	0.40
PCA + KNN	1.00	0.41	0.41	0.41
Naïve bayes	0.4943	0.47	0.45	0.44
PCA + Naïve bayes	0.4943	0.54	0.48	0.45
Regresión Log.	0.4968	0.49	0.48	0.48
PCA + Regresión Log.	0.5093	0.55	0.52	0.52
SVM	1.00	0.27	0.35	0.23
PCA + SVM	1.00	0.31	0.36	0.23

Experimento 6

Ya que los resultados en el experimento 5 no fueron satisfactorios, en este experimento se realizaron tres combinaciones entre las clases 52 (eclampsia durante el trabajo de parto), 54 (eclampsia, en período no especificado) y 163 (hemorragia postparto secundaria o tardía). Estos experimentos llevan por nombre 6A, 6B y 6C y se presentan a continuación.

Experimento 6A (combinación de las clases 52 y 54)

En la Tabla 8 se muestran los resultados al utilizar las instancias de las clases 52 (eclampsia durante el trabajo de parto) y 54 (eclampsia, en período no especificado), con un total de 1,157 instancias.

Tabla 8

Resultados de la aplicación del grupo de algoritmos al conjunto de datos que contiene las clases 52 y 54

Algoritmo Clasificador	Accuracy	Precision	Recall	F1-Score
KNN	1.00	0.52	0.52	0.52
PCA + KNN	1.00	0.50	0.50	0.50
Naïve bayes	0.5695	0.56	0.51	0.47
PCA + Naïve bayes	0.5695	0.56	0.56	0.53
Regresión Log.	0.5911	0.55	0.55	0.55
PCA + Regresión Log.	0.5980	0.59	0.59	0.59
SVM	1.00	0.49	0.50	0.46
PCA + SVM	1.00	0.42	0.47	0.34

Experimento 6B (combinación de las clases 54 y 163)

En la Tabla 9 se muestran los resultados al utilizar las instancias de las clases 54 (eclampsia en período no especificado) y 163 (hemorragia postparto secundaria o tardía), con un total de 1,025 instancias

Experimento 6C (combinación de las clases 52 y 163)

En la Tabla 10 se muestran los resultados al utilizar las instancias de las clases 52 (eclampsia durante el trabajo de parto) y 163 (hemorragia postparto secundaria o tardía), con un total de 1,018 instancias. En esta tabla se puede observar un incremento, tanto en accuracy como en precision y recall, de varios algoritmos. Estos resultados superan a los encontrados en los experimentos 6A y 6B.

Las figuras que se muestran a continuación, pertenecen a los tres algoritmos que mejor clasifican en términos de alta precision y recall en la Tabla 10: naïve bayes (ver

Figura 13), PCA + naïve bayes (ver Figura 14), regresión logística (ver Figura 15), PCA + regresión logística (ver Figura 16). La decisión de tomar los valores más altos de precision y recall se basó en contar con un bajo número de falsos positivos y de falsos negativos.

Tabla 9

Resultados de la aplicación del grupo de algoritmos al conjunto de datos que contiene las clases 54 y 163

Algoritmo Clasificador	Accuracy	Precision	Recall	F1-Score
KNN	1.0	0.45	0.55	0.39
PCA + KNN	1.0	0.60	0.61	0.60
Naïve bayes	0.6985	0.74	0.72	0.70
PCA + naïve bayes	0.6985	0.75	0.71	0.69
Regresión Log.	0.6858	0.70	0.70	0.70
PCA + regresión log.	0.6936	0.71	0.71	0.70
SVM	1.0	0.32	0.56	0.40
PCA + SVM	1.0	0.34	0.58	0.43

Tabla 10

Resultados de la aplicación del grupo de algoritmos al conjunto de datos que contiene las clases 52 y 163

Algoritmo Clasificador	Accuracy	Precision	Recall	F1-Score
KNN	1.00	0.60	0.59	0.60
PCA + KNN	1.00	0.68	0.68	0.68
Naïve bayes	0.7236	0.75	0.74	0.71
PCA + Naïve bayes	0.7217	0.77	0.71	0.69
Regresión Log.	0.7315	0.73	0.73	0.73
PCA + Regresión Log.	0.7345	0.72	0.72	0.72
SVM	1.00	0.54	0.57	0.42
PCA + SVM	1.00	0.31	0.56	0.40

```
>>> #Nabie Bayes muerteBCeros52,163.txt
... print(classification_report(y_test, y_pred, target_names=target_names))
             precision    recall  f1-score   support

         52       0.71      0.94      0.81       185
        163       0.81      0.43      0.56       121

avg / total       0.75      0.74      0.71       306

>>>
>>> accuracy = nb.score(X, y)
>>> print(accuracy)
0.723697148476
```

Figura 13. Resultado obtenido en el experimento con las clases 52 y 163 con el algoritmo naïve bayes.

```
>>> #Nave Bayes + PCA muerteBCeros52,163.txt
... print(classification_report(y_test, y_pred, target_names=target_names))
             precision    recall  f1-score   support

         52       0.66      0.96      0.78       164
        163       0.90      0.43      0.58       142

avg / total       0.77      0.71      0.69       306

>>>
>>> accuracy = nb.score(X, y)
>>> print(accuracy)
0.721730580138
```

Figura 14. Resultado obtenido en el experimento con las clases 52 y 163 con el algoritmo naïve bayes + PCA.

```
>>> #Log. Regression muerteBCeros52,163.txt
... print(classification_report(y_test, y_pred, target_names=target_names))
             precision    recall  f1-score   support

         52       0.74      0.80      0.77       172
        163       0.71      0.63      0.67       134

avg / total       0.73      0.73      0.73       306

>>>
>>> accuracy = logreg.score(X, y)
>>> print(accuracy)
0.731563421829
```

Figura 15. Resultado obtenido en el experimento con las clases 52 y 163 con el algoritmo regresión logística.

```
>>> #Log. Regression + PCA muerteBCeros52,163.txt
... print(classification_report(y_test, y_pred, target_names=target_names))
             precision    recall  f1-score   support

         52       0.72      0.84      0.78       175
        163       0.73      0.56      0.64       131

avg / total       0.72      0.72      0.72       306

>>>
>>> accuracy = logistic.score(X_digits, y_digits)
>>> print(accuracy)
0.734513274336
```

Figura 16. Resultado obtenido en el experimento con las clases 52 y 163 con el algoritmo regresión logística + PCA.

CAPÍTULO V

DISCUSIÓN, CONCLUSIONES Y TRABAJO FUTURO

Discusión

Como se demuestra en la Tabla 10 y de acuerdo con el experimento 6C, es posible realizar una clasificación efectiva de las clases 52 (eclampsia durante el trabajo de parto) con 576 instancias y 163 (hemorragia postparto secundaria o tardía) con 442 instancias, con un total de 1,018 registros.

De acuerdo a la Tabla 10, los algoritmos con mejores resultados en términos de f1-score son naïve bayes y regresión logística. Los valores de precision y recall en el caso de estos dos clasificadores son muy similares. Es por esto que cualquiera de ellos se hubiera podido utilizar para generar el modelo que se utiliza por el software creado. Sin embargo, se decidió utilizar el algoritmo de naïve bayes porque este es el que presenta en general una mayor precision (menor número de falsos positivos) y un mayor recall (menor número de falsos negativos).

En otras palabras, es posible clasificar con un accuracy de 0.7236 a las mujeres en control de embarazo, derivadas de las causas de muerte por eclampsia durante el trabajo de parto (precision de 0.71, recall de 0.94 y f1-score de 0.81) o de padecer hemorragia postparto secundaria o tardía (precision de 0.81, recall de 0.43 y f1-score de 0.56). Aunque el resultado del recall en el caso de eclampsia durante el trabajo de parto es bajo (0.43), este resultado es útil en este caso, pues el interés de este estudio

está enfocado más hacia la precision que hacia el recall.

El modelo entrenado mediante naïve bayes con las instancias de las clases 52 y 163 es utilizado para predecir si alguna madre se puede clasificar en alguna de estas dos posibles causas de muerte. Con este fin, el personal médico ingresa los siguientes datos: mes de nacimiento, día de nacimiento, edad cumplida, estado conyugal, entidad de residencia, tamaño de localidad, ocupación habitual, escolaridad, derechohabiencia, entidad, municipio, localidad y asistencia médica (ver el Apéndice A en donde se muestran los tipos de datos y la explicación de cada una de las características). Estos datos son utilizados para mostrar los resultados de la inferencia con base en el modelo entrenado de naïve bayes.

Conclusiones

En México, la Dirección General de Información en Salud (2016) maneja una base de datos abierta sobre muerte materna que fue creada con datos recopilados de cada estado de la República Mexicana por parte de la Secretaría de Salud. Esta base de datos está organizada en 58 variables o características. Dentro de estas variables, se encuentra la variable “causa de clasificación internacional de enfermedades (CIE)” que contiene un total de 248 enfermedades que están relacionadas con muertes maternas.

En esta investigación se construyó un software para la clasificación de riesgos de la muerte materna en México mediante la aplicación de ciencia de datos y aprendizaje automático.

Con el fin de construir este software, en este trabajo se descubrieron las variables que pueden ser usadas para realizar una clasificación, mediante la metodología de ciencia de datos de IBM (Rollins, 2015) y de aprendizaje automático (en una forma

retrospectiva) para clasificar a pacientes embarazadas con riesgo de mortalidad.

Específicamente, el modelo utilizado por el software fue el generado con el algoritmo de aprendizaje supervisado naïve bayes, con el cual se obtuvo un accuracy de 0.7236, una precision general de 0.75, un recall general de 0.74, un f1-score general de 0.71 y para las clases eclampsia durante el trabajo de parto (precisión de 0.71, recall de 0.94 y f1-score de 0.81) y hemorragia postparto secundaria o tardía (precisión de 0.81, recall de 0.43 y f1-score de 0.56). Este modelo predictivo se ejecuta cada vez que un médico introduce los datos de alguna paciente en el sistema.

Las contribuciones particulares de esta investigación son las siguientes:

1. Se redujo el número de variables de riesgo de muerte materna mediante el uso de ciencia de datos y de aprendizaje automático.

2. Se crearon modelos predictivos para la clasificación de enfermedades que pueden llevar a la muerte materna, mediante aprendizaje automático.

3. Se evaluaron los modelos de clasificación generados.

4. Se creó un software que utiliza el modelo predictivo con los mejores resultados para clasificar a mujeres embarazadas de acuerdo con los riesgos de mortalidad.

Trabajos futuros

A futuro se espera realizar una serie de investigaciones considerando lo siguiente:

1. Se espera que esta investigación pueda ser ampliada mediante la utilización de un conjunto de datos más actualizado de la Dirección General de Información en Salud (2016). Esto, con el fin de contar con un mayor número de instancias que puedan permitir obtener mejores resultados.

2. Se espera desplegar el prototipo de software desarrollado en ambientes reales con el fin de obtener resultados adicionales, tales como la comparación entre los resultados de la herramienta versus la clasificación de pacientes con base en el dictamen médico.

3. Se espera extender el programa para no solamente clasificar a pacientes nuevos sino para guardar su información para así extender el conjunto de datos que se pueden utilizar para crear futuros modelos predictivos.

4. Se espera ampliar la base de datos utilizada en los experimentos con otros conjuntos de datos abiertos. Esto con la finalidad de tener más características disponibles, tales como la fecha esperada de nacimiento, con las cuales se puedan obtener modelos más sólidos.

APÉNDICE A

CARACTERÍSTICAS DEL CONJUNTO DE DATOS DE MUERTE MATERNA

Núm.	Características	Tipo de dato
1	Año de nacimiento	Integer
2	Mes de nacimiento	Integer
3	Descripción del mes de nacimiento	String
4	Día de nacimiento	Integer
5	Genero	Integer
6	Clave de la edad	String
7	Edad cumplida	Integer
8	Estado conyugal	Integer
9	Descripción del estado conyugal	String
10	Entidad de residencia	Integer
11	Descripción de entidad de residencia	String
12	Municipio de residencia	Integer
13	Descripción del municipio de residencia	String
14	Localidad de residencia	Integer
15	Descripción de la localidad de residencia	String
16	Tamaño de localidad	Integer
17	Descripción del tamaño de localidad	String
18	Ocupación habitual	Integer
19	Descripción de la ocupación habitual	String
20	Escolaridad	Integer
21	Descripción de la escolaridad	String
22	Derechohabiencia	Integer
23	Descripción de la derechohabiencia	String
24	Entidad de ocurrencia	Integer
25	Descripción de la entidad de ocurrencia	String
26	Municipio de ocurrencia	Integer
27	Descripción del municipio de ocurrencia	String
28	Localidad de ocurrencia	Integer
29	Descripción de la localidad de ocurrencia	String
30	Sitio donde ocurrió la defunción	Integer
31	Descripción del sitio donde ocurrió la defunción	String
32	Año de la defunción	Integer
33	Mes de la defunción	Integer
34	Descripción del mes de la defunción	String
35	Día de la defunción	Integer
36	Hora de la defunción	Integer
37	Minutos de la defunción	Integer
38	Asistencia médica	Integer

39	Descripción de la asistencia médica	String
40	Causa CIE a cuarto dígito	Integer
41	Descripción de la causa CIE	String
42	Quien certificó	Integer
43	Descripción de quien certificó	String
44	Entidad de registro	Integer
45	Descripción de la entidad de registro	String
46	Municipio de registro	Integer
47	Descripción del municipio de registro	String
48	Año de registro	Integer
49	Mes de registro	Integer
50	Descripción del mes de registro	String
51	Día de registro	Integer
52	Año de la certificación	Integer
53	Mes de la certificación	Integer
54	Descripción del mes de la certificación	String
55	Día de la certificación	Integer
56	Año de la base de datos	Integer
57	Razón de mortalidad materna	Integer
58	Descripción de la razón de mortalidad materna	String

APÉNDICE B

DESCRIPCIÓN DE LAS CLASES

Núm.	Clases determinadas por CIE a cuatro dígitos, mas una descripción
1	A34X Tétanos obstétrico
2	B200 Enfermedad por VIH, resultante en infección por micro bacterias
3	O000 Embarazo abdominal
4	O001 Embarazo tubárico
5	O002 Embarazo ovárico
6	O008 Otros embarazos ectópicos
7	O009 Embarazo ectópico, no especificado
8	O010 Mola hidatiforme clásica
9	O011 Mola hidatiforme, incompleta o parcial
10	O019 Mola hidatiforme, no especificada
11	O020 Detención del desarrollo del huevo y mola no hidatiforme
12	O021 Aborto retenido
13	O030 Aborto espontáneo incompleto, complicado con infección genital y pelviana
14	O031 Aborto espontáneo incompleto, complicado por hemorragia excesiva o tardía
15	O032 Aborto espontáneo incompleto, complicado por embolia
16	O033 Aborto espontáneo incompleto, con otras complicaciones especificadas y las no especificadas
17	O034 Aborto espontáneo incompleto, sin complicación
18	O035 Aborto espontáneo completo o no especificado, complicado con infección genital y pelviana
19	O036 Aborto espontáneo completo o no especificado, complicado por hemorragia excesiva o tardía
20	O038 Aborto espontáneo completo o no especificado, con otras complicaciones especificadas y las no especificadas
21	O039 Aborto espontáneo completo o no especificado, sin complicación
22	O040 Aborto médico incompleto, complicado con infección genital y pelviana
23	O041 Aborto médico incompleto, complicado por hemorragia excesiva o tardía
24	O045 Aborto médico completo o no especificado, complicado con infección genital y pelviana
25	O048 Aborto médico completo o no especificado, con otras complicaciones especificadas y las no especificadas
26	O050 Otro aborto incompleto, complicado con infección genital y pelviana
27	O053 Otro aborto incompleto, con otras complicaciones especificadas y las no especificadas
28	O056 Otro aborto completo o no especificado, complicado por hemorragia excesiva o tardía
29	O058 Otro aborto completo o no especificado, con otras complicaciones especificadas y las no especificadas

30	O059 Otro aborto completo o no especificado, sin complicación
31	O060 Aborto no especificado incompleto, complicado con infección genital y pelviana
32	O061 Aborto no especificado incompleto, complicado por hemorragia excesiva o tardía
33	O062 Aborto no especificado incompleto, complicado por embolia
34	O063 Aborto no especificado incompleto, con otras complicaciones especificadas y las no especificadas
35	O064 Aborto no especificado incompleto, sin complicación
36	O065 Aborto no especificado completo o no especificado, complicado con infección genital y pelviana
37	O066 Aborto no especificado completo o no especificado, complicado por hemorragia excesiva o tardía
38	O067 Aborto no especificado completo o no especificado, complicado por embolia
39	O068 Aborto no especificado completo o no especificado, con otras complicaciones especificadas y las no especificadas
40	O069 Aborto no especificado completo o no especificado, sin complicación
41	O100 Hipertensión esencial preexistente que complica el embarazo, el parto y el puerperio
42	O101 Enfermedad cardíaca hipertensiva preexistente que complica el embarazo, el parto y el puerperio
43	O102 Enfermedad renal hipertensiva preexistente que complica el embarazo, el parto y el puerperio
44	O109 Hipertensión preexistente no especificada, que complica el embarazo, el parto y el puerperio
45	O11X Trastornos hipertensivos preexistentes, con proteinuria agregada
46	O120 Edema gestacional
47	O13X Hipertensión gestacional [inducida por el embarazo] sin proteinuria significativa
48	O140 Preeclampsia moderada
49	O141 Preeclampsia severa
50	O149 Preeclampsia, no especificada
51	O150 Eclampsia en el embarazo
52	O151 Eclampsia durante el trabajo de parto
53	O152 Eclampsia en el puerperio
54	O159 Eclampsia, en período no especificado
55	O16X Hipertensión materna, no especificada
56	O210 Hiperemesis gravídica leve
57	O211 Hiperemesis gravídica con trastornos metabólicos
58	O220 Venas varicosas de los miembros inferiores en el embarazo
59	O222 Tromboflebitis superficial en el embarazo

60	O223 Flebotrombosis profunda en el embarazo
61	O225 Trombosis venosa cerebral en el embarazo
62	O229 Complicación venosa no especificada en el embarazo
63	O230 Infección del riñón en el embarazo
64	O233 Infección de otras partes de las vías urinarias en el embarazo
65	O234 Infección no especificada de las vías urinarias en el embarazo
66	O235 Infección genital en el embarazo
67	O240 Diabetes mellitus preexistente insulinodependiente, en el embarazo
68	O241 Diabetes mellitus preexistente no insulinodependiente, en el embarazo
69	O242 Diabetes mellitus preexistente relacionada con desnutrición, en el embarazo
70	O243 Diabetes mellitus preexistente, sin otra especificación, en el embarazo
71	O244 Diabetes mellitus que se origina con el embarazo
72	O249 Diabetes mellitus no especificada, en el embarazo
73	O25X Desnutrición en el embarazo
74	O265 Síndrome de hipotensión materna
75	O266 Trastornos del hígado en el embarazo, el parto y el puerperio
76	O268 Otras complicaciones especificadas relacionadas con el embarazo
77	O269 Complicación relacionada con el embarazo, no especificada
78	O290 Complicaciones pulmonares de la anestesia administrada durante el embarazo
79	O291 Complicaciones cardíacas de la anestesia administrada durante el embarazo
80	O292 Complicaciones del sistema nervioso central debidas a la anestesia administrada durante el embarazo
81	O293 Reacción tóxica a la anestesia local administrada durante el embarazo
82	O295 Otras complicaciones de la anestesia espinal o epidural administradas durante el embarazo
83	O298 Otras complicaciones de la anestesia administrada durante el embarazo
84	O300 Embarazo doble
85	O301 Embarazo triple
86	O321 Atención materna por presentación de nalgas
87	O322 Atención materna por posición fetal oblicua o transversa
88	O326 Atención materna por presentación compuesta
89	O331 Atención materna por desproporción debida a estrechez general de la pelvis
90	O335 Atención materna por desproporción debida a feto demasiado grande

91	O339 Atención materna por desproporción de origen no especificado
92	O340 Atención materna por anomalía congénita del útero
93	O341 Atención materna por tumor del cuerpo del útero
94	O342 Atención materna por cicatriz uterina debida a cirugía previa
95	O343 Atención materna por incompetencia del cuello uterino
96	O345 Atención materna por otras anormalidades del útero grávido
97	O348 Atención materna por otras anormalidades de los órganos pelvianos
98	O350 Atención materna por (presunta) malformación del sistema nervioso central en el feto
99	O359 Atención materna por (presunta) anormalidad y lesión fetal no especificada
100	O363 Atención materna por signos de hipoxia fetal
101	O364 Atención materna por muerte intrauterina
102	O366 Atención materna por crecimiento fetal excesivo
103	O367 Atención materna por feto viable en embarazo abdominal
104	O368 Atención materna por otros problemas fetales especificados
105	O40X Polihidramnios
106	O410 Oligohidramnios
107	O411 Infección de la bolsa amniótica o de las membranas
108	O420 Ruptura prematura de las membranas, e inicio del trabajo de parto dentro de las 24 horas
109	O421 Ruptura prematura de las membranas, e inicio del trabajo de parto después de las 24 horas
110	O429 Ruptura prematura de las membranas, sin otra especificación
111	O438 Otros trastornos placentarios
112	O439 Trastorno de la placenta, no especificado
113	O440 Placenta previa con especificación de que no hubo hemorragia
114	O441 Placenta previa con hemorragia
115	O450 Desprendimiento prematuro de la placenta con defecto de la coagulación
116	O458 Otros desprendimientos prematuros de la placenta
117	O459 Desprendimiento prematuro de la placenta, sin otra especificación
118	O460 Hemorragia anteparto con defecto de la coagulación
119	O468 Otras hemorragias anteparto
120	O469 Hemorragia anteparto, no especificada
121	O470 Falso trabajo de parto antes de las 37 semanas completas de gestación
122	O48X Embarazo prolongado
123	O60X Parto prematuro
124	O620 Contracciones primarias inadecuadas
125	O622 Otras inercias uterinas

126	O624 Contracciones uterinas hipertónicas, incoordinadas y prolongadas
127	O629 Anomalía dinámica del trabajo de parto, no especificada
128	O631 Prolongación del segundo período (del trabajo de parto) O Prolongación del primer período (del trabajo de parto)
129	O639 Trabajo de parto prolongado, no especificado
130	O641 Trabajo de parto obstruido debido a presentación de nalgas
131	O642 Trabajo de parto obstruido debido a presentación de cara
132	O644 Trabajo de parto obstruido debido a presentación de hombro
133	O648 Trabajo de parto obstruido debido a otras presentaciones anormales del feto
134	O649 Trabajo de parto obstruido debido a presentación anormal del feto no especificada
135	O654 Trabajo de parto obstruido debido a desproporción fetopelviana, sin otra especificación
136	O655 Trabajo de parto obstruido debido a anomalías de los órganos pelvianos maternos
137	O660 Trabajo de parto obstruido debido a distocia de hombros
138	O661 Trabajo de parto obstruido debido a distocia gemelar
139	O662 Trabajo de parto obstruido debido a distocia por feto inusualmente grande
140	O663 Trabajo de parto obstruido debido a otras anormalidades del feto
141	O668 Otras obstrucciones especificadas del trabajo de parto
142	O669 Trabajo de parto obstruido, sin otra especificación
143	O679 Hemorragia intraparto, no especificada O Otras hemorragias intraparto
144	O689 Trabajo de parto y parto complicados por sufrimiento fetal, sin otra especificación
145	O690 Trabajo de parto y parto complicados por prolapso del cordón umbilical
146	O691 Trabajo de parto y parto complicados por circular pericervical del cordón, con compresión
147	O699 Trabajo de parto y parto complicados por problemas no especificados del cordón umbilical
148	O700 Desgarro perineal de primer grado durante el parto
149	O702 Desgarro perineal de tercer grado durante el parto
150	O703 Desgarro perineal de cuarto grado durante el parto
151	O710 Ruptura del útero antes del inicio del trabajo de parto
152	O711 Ruptura del útero durante el trabajo de parto
153	O712 Inversión del útero, postparto
154	O713 Desgarro obstétrico del cuello uterino
155	O714 Desgarro vaginal obstétrico alto, sólo
156	O715 Otros traumatismos obstétricos de los órganos pelvianos
157	O716 Traumatismo obstétrico de los ligamentos y articulaciones de la pelvis
158	O717 Hematoma obstétrico de la pelvis

159	O718 Otros traumas obstétricos especificados
160	O719 Trauma obstétrico, no especificado
161	O720 Hemorragia del tercer período del parto
162	O721 Otras hemorragias postparto inmediatas
163	O722 Hemorragia postparto secundaria o tardía
164	O723 Defecto de la coagulación postparto
165	O730 Retención de la placenta sin hemorragia
166	O731 Retención de fragmentos de la placenta o de las membranas, sin hemorragia
167	O740 Neumonitis por aspiración debida a la anestesia administrada durante el trabajo de parto y el parto
168	O742 Complicaciones cardíacas de la anestesia administrada durante el trabajo de parto y el parto
169	O743 Complicaciones del sistema nervioso central por la anestesia administrada durante el trabajo de parto y el parto
170	O744 Reacción tóxica a la anestesia local administrada durante el trabajo de parto y el parto
171	O745 Cefalalgia inducida por la anestesia espinal o epidural administradas durante el trabajo de parto y el parto
172	O746 Otras complicaciones de la anestesia espinal o epidural administradas durante el trabajo de parto y el parto
173	O747 Falla o dificultad en la intubación durante el trabajo de parto y el parto
174	O748 Otras complicaciones de la anestesia administrada durante el trabajo de parto y el parto
175	O749 Complicación no especificada de la anestesia administrada durante el trabajo de parto y el parto
176	O751 Choque durante o después del trabajo de parto y el parto
177	O753 Otras infecciones durante el trabajo de parto
178	O754 Otras complicaciones de la cirugía y otros procedimientos obstétricos
179	O758 Otras complicaciones especificadas del trabajo de parto y del parto
180	O759 Complicación no especificada del trabajo de parto y del parto
181	O85X Sepsis puerperal
182	O860 Infección de herida quirúrgica obstétrica
183	O864 Pirexia de origen desconocido consecutiva al parto
184	O862 Infección de las vías urinarias consecutiva al parto
185	O861 Otras infecciones genitales consecutivas al parto
186	O868 Otras infecciones puerperales especificadas
187	O870 Tromboflebitis superficial en el puerperio
188	O871 Flebotrombosis profunda en el puerperio
189	O873 Trombosis venosa cerebral en el puerperio
190	O878 Otras complicaciones venosas en el puerperio
191	O879 Complicación venosa en el puerperio, no especificada

192	O880 Embolia gaseosa, obstétrica
193	O881 Embolia de líquido amniótico
194	O882 Embolia de coágulo sanguíneo, obstétrica
195	O883 Embolia séptica y piémica, obstétrica
196	O888 Otras embolias obstétricas
197	O891 Complicaciones cardíacas de la anestesia administrada durante el puerperio
198	O892 Complicaciones del sistema nervioso central debidas a la anestesia administrada durante el puerperio
199	O893 Reacción tóxica a la anestesia local administrada durante el puerperio
200	O895 Otras complicaciones de la anestesia espinal o epidural administradas durante el puerperio
201	O898 Otras complicaciones de la anestesia administrada durante el puerperio
202	O900 Dehiscencia de sutura de cesárea
203	O901 Dehiscencia de sutura obstétrica perineal
204	O902 Hematoma de herida quirúrgica obstétrica
205	O903 Cardiomiopatía en el puerperio
206	O904 Insuficiencia renal aguda postparto
207	O908 Otras complicaciones puerperales, no clasificadas en otra parte
208	O909 Complicación puerperal, no especificada
209	O912 Mastitis no purulenta asociada con el parto
210	O95X Muerte obstétrica de causa no especificada
211	O96X Muerte materna debida a cualquier causa obstétrica que ocurre después de 42 días pero antes de un año del parto
212	O97X Muerte por secuelas de causas obstétricas directas
213	O980 Tuberculosis que complica el embarazo, el parto y el puerperio
214	O983 Otras infecciones con un modo de transmisión predominantemente sexual que complican el embarazo, parto y puerperio
215	O984 Hepatitis viral que complica el embarazo, el parto y el puerperio
216	O985 Otras enfermedades virales que complican el embarazo, el parto y el puerperio
217	O986 Enfermedades causadas por protozoarios que complican el embarazo, el parto y el puerperio
218	O988 Otras enfermedades infecciosas y parasitarias maternas que complican el embarazo, el parto y el puerperio
219	O989 Enfermedad infecciosa y parasitaria materna no especificada que complica el embarazo, el parto y el puerperio
220	O990 Anemia que complica el embarazo, el parto y el puerperio
221	O991 Otras enfermedades de la sangre y órganos hematopoyéticos y ciertos trastornos que afectan el sistema inmunitario cuando complican el embarazo, el parto y puerperio

222	O992 Enfermedades endocrinas, de la nutrición y del metabolismo que complican el embarazo, el parto y el puerperio
223	O993 Trastornos mentales y enfermedades del sistema nervioso que complican el embarazo, el parto y el puerperio
224	O994 Enfermedades del sistema circulatorio que complican el embarazo, el parto y el puerperio
225	O995 Enfermedades del sistema respiratorio que complican el embarazo, el parto y el puerperio
226	O996 Enfermedades del sistema digestivo que complican el embarazo, el parto y el puerperio
227	O997 Enfermedades de la piel y del tejido subcutáneo que complican el embarazo, el parto y el puerperio
228	O998 Otras enfermedades especificadas y afecciones que complican el embarazo, el parto y el puerperio
229	F530 Trastornos mentales y del comportamiento leves, asociados con el puerperio, no clasificados en otra parte
230	F531 Trastornos mentales y del comportamiento graves, asociados con el puerperio, no clasificados en otra parte
231	D392 Tumor de comportamiento incierto o desconocido de la placenta
232	C58X Tumor maligno de la placenta
233	B201 Enfermedad por VIH, resultante en otras infecciones bacterianas
234	B203 Enfermedad por VIH, resultante en otras infecciones virales
235	B204 Enfermedad por VIH, resultante en candidiasis
236	B205 Enfermedad por VIH, resultante en otras micosis
237	B206 Enfermedad por VIH, resultante en neumonía por Pneumocystis carinii
238	B207 Enfermedad por VIH, resultante en infecciones múltiples
239	B208 Enfermedad por VIH, resultante en otras enfermedades infecciosas o parasitarias
240	B209 Enfermedad por VIH, resultante en enfermedad infecciosa o parasitaria no especificada
241	B218 Enfermedad por VIH, resultante en otros tumores malignos
242	B220 Enfermedad por VIH, resultante en encefalopatía
243	B222 Enfermedad por VIH, resultante en síndrome caquéctico
244	B227 Enfermedad por VIH, resultante en enfermedades múltiples clasificadas en otra parte
245	B230 Síndrome de infección aguda debida a VIH
246	B232 Enfermedad por VIH, resultante en anormalidades inmunológicas y hematológicas, no clasificadas en otra parte
247	B238 Enfermedad por VIH, resultante en otras afecciones especificadas
248	B24X Enfermedad por virus de la inmunodeficiencia humana [VIH], sin otra especificación

APÉNDICE C

CÓDIGOS DE LA INTERFAZ DE USUARIO

Form.html y Custom.js

La interfaz de usuario se compone de Form.html y Custom.js. Debido a lo extenso de estos archivos, se comparte mediante Google drive en la siguiente dirección web de la cual se puede descargar.

https://drive.google.com/drive/folders/0BzO8zQQuN9yjZGlOSkp3bU50NGM?usp=sharing

Proccesnp.php

```
<?php

header('Access-Control-Allow-Origin: *');

$salida = system('Python C:\Users\Ruxbel\Anaconda2\Lib\site-packages\PyQt4\EjecutaModelo.py '.$_REQUEST["uno"]." ".$_REQUEST["dos"]." ".$_REQUEST["tres"]." ".$_REQUEST["cuatro"]." ".$_REQUEST["cinco"]." ".$_REQUEST["seis"]." ".$_REQUEST["siete"]." ".$_REQUEST["ocho"]." ".$_REQUEST["nueve"]." ".$_REQUEST["diez"]." ".$_REQUEST["once"]." ".$_REQUEST["doce"]." ".$_REQUEST["trece"],$response);

//echo $salida;

 ?>
```

EjecutaModelo.py

```
from sklearn.externals import joblib
from sklearn.naive_bayes import GaussianNB
from sklearn.cross_validation import train_test_split
from sklearn import cross_validation
import pickle
import csv, operator
import pandas as pd
import numpy as np
import sys
import warnings

warnings.filterwarnings("ignore", category=DeprecationWarning)

df = pd.read_csv('C:\Users\Ruxbel\Anaconda2\Lib\site-packages\PyQt4\Clases52,163.txt')
X = np.array(df.drop(['14_Causa_CIE_a_Cuarto_digito'], 1))
y = np.array(df['14_Causa_CIE_a_Cuarto_digito'])
```

```
X_train, X_test, y_train, y_test = cross_validation.train_test_split(X, y,
test_size=0.3)
nb = GaussianNB()
nb.fit(X_train, y_train)

y_pred = nb.fit(X=X_train, y=y_train).predict(X_test);
nb = GaussianNB()
joblib.dump(nb, 'C:\Users\Ruxbel\Anaconda2\Lib\site-packages\PyQt4\\ModeloNB.pkl')
nb = joblib.load('C:\Users\Ruxbel\Anaconda2\Lib\site-pack-
ages\PyQt4\\ModeloNB.pkl')
nb.fit(X=X_train, y=y_train).predict(X_test)

reg2 = []
reg2.append(len(sys.argv[0]))
reg2.append(len(sys.argv[1]))
reg2.append(len(sys.argv[2]))
reg2.append(len(sys.argv[3]))
reg2.append(len(sys.argv[4]))
reg2.append(len(sys.argv[5]))
reg2.append(len(sys.argv[6]))
reg2.append(len(sys.argv[7]))
reg2.append(len(sys.argv[8]))
reg2.append(len(sys.argv[9]))
reg2.append(len(sys.argv[10]))
reg2.append(len(sys.argv[11]))
reg2.append(len(sys.argv[12]))

if nb.predict(reg2) == 52:
       print "La paciente puede padecer de eclampsia durante el trabajo de parto"
if nb.predict(reg2) == 163:
       print "La paciente puede padecer de emorragia postparto secundaria o tardia"
```

APÉNDICE D

CÓDIGO DE LOS EXPERIMENTOS

KNN

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import preprocessing, cross_validation, neighbors
import pandas as pd
from sklearn import svm, datasets
from sklearn.cross_validation import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn import cross_validation
from sklearn.metrics import classification_report
import os
#os.system("cls")

#Importando Dataset----------------------

df = pd.read_csv('muerte.txt')
X = np.array(df.drop(['14_Causa_CIE_a_Cuarto_digito'], 1))
y = np.array(df['14_Causa_CIE_a_Cuarto_digito'])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
#test_size=0.3

#Aplicando el Clasificador-----------------
#visite la siguiente dirección web para conocer los detalles de los siguientes
#parametro
#http://scikitlearn.org/stable/modules/genera-
ted/sklearn.neighbors.KNeighborsCl#assifier.html

classifier = neighbors.KNeighborsClassifier(algorithm='brute', leaf_size=30, met-
ric='minkowski',
 metric_params=None, n_jobs=1, n_neighbors=3, p=1,
 weights='distance')
y_pred = classifier.fit(X_train, y_train).predict(X_test)

classifier.score(X_test, y_test)
print(classifier.score(X,y))

print 'score = ', classifier.score(X_test, y_test)

#Realizando el Reporte-------------------------------

print y_pred[0:20]
print y_test[0:20]

target_names =
['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18',
'19','20','21','22','23','24','26','27','28','29','30','31','32','33','34','35','3
6','37','38','39','40','41','42','43','44','45','46','47','48','49','50','51','52'
,'53','54','55','56','57','58','59','60','61','62','63','64','65','66','67','68','
69','70','71','72','73','74','75','76','77','78','79','80','81','82','83','84','85
','86','87','88','89','90','91','92','93','94','95','96','97','98','99','100','101
','102','103','104','105','106','107','108','109','110','111','112','113','114','1
15','116','117','118','119','120','121','122','123','124','125','126','127','128',
```

```
'129','130','131','132','133','134','135','136','137','138','139','140','141','142
','143','144','145','146','147','148','149','150','151','152','153','154','155','1
56','157','158','159','160','161','162','163','164','165','166','167','168','169',
'171','172','173','174','175','176','177','178','179','180','181','182','183','184
','185','186','187','188','189','190','191','192','193','194','195','196','197','1
98','199','200','201','202','203','204','205','206','207','208','209','210','211',
'212','213','214','215','216','217','218','219','220','221','222','223','224','225
','226','227','228','229','230','231','232','233','234','235','236','237','238','2
39','240','241','242','243','244','245','24','247']

print (classification_report(y_test, y_pred, target_names=target_names))

#Resultados prebios-------------------------------------

accuracy = classifier.score(X, y)
classifier.score(X_test, y_test)
print(classifier.predict(X))
print(accuracy)
```

KNN + PCA

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import preprocessing, cross_validation, neighbors
import pandas as pd
from sklearn import svm, datasets
from sklearn.cross_validation import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.decomposition import PCA
from sklearn import cross_validation
import os
from sklearn.metrics import classification_report
#os.system("cls")

#Importando Dataset----------------------

df = pd.read_csv('muerte.txt')
X = np.array(df.drop(['14_Causa_CIE_a_Cuarto_digito'], 1))
y = np.array(df['14_Causa_CIE_a_Cuarto_digito'])

#Aplicando PCA---------------------------

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
pca = PCA(n_components=2)
pca.fit(X_train)
X_t_train = pca.transform(X_train)
X_t_test = pca.transform(X_test)

pca.fit_transform(X [y])

#Aplicando el Clasificador------------------
```

```
classifier = neighbors.KNeighborsClassifier(algorithm='brute', leaf_size=30, met-
ric='minkowski',
 metric_params=None, n_jobs=1, n_neighbors=3, p=1,
 weights='distance')
y_pred = classifier.fit(X_train, y_train).predict(X_test)

#Realizando el Reporte--------------------------------

print y_pred[0:20]
print y_test[0:20]

target_names =
['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18',
'19','20','21','22','23','24','26','27','28','29','30','31','32','33','34','35','3
6','37','38','39','40','41','42','43','44','45','46','47','48','49','50','51','52'
,'53','54','55','56','57','58','59','60','61','62','63','64','65','66','67','68','
69','70','71','72','73','74','75','76','77','78','79','80','81','82','83','84','85
','86','87','88','89','90','91','92','93','94','95','96','97','98','99','100','101
','102','103','104','105','106','107','108','109','110','111','112','113','114','1
15','116','117','118','119','120','121','122','123','124','125','126','127','128',
'129','130','131','132','133','134','135','136','137','138','139','140','141','142
','143','144','145','146','147','148','149','150','151','152','153','154','155','1
56','157','158','159','160','161','162','163','164','165','166','167','168','169',
'171','172','173','174','175','176','177','178','179','180','181','182','183','184
','185','186','187','188','189','190','191','192','193','194','195','196','197','1
98','199','200','201','202','203','204','205','206','207','208','209','210','211',
'212','213','214','215','216','217','218','219','220','221','222','223','224','225
','226','227','228','229','230','231','232','233','234','235','236','237','238','2
39','240','241','242','243','244','245','24','247']

print (classification_report(y_test, y_pred, target_names=target_names))

classifier.score(X_test, y_test)
print(classifier.score(X,y))

print 'score = ', classifier.score(X_test, y_test)

########################################
accuracy = classifier.score(X, y)
classifier.score(X_test, y_test)
print(classifier.predict(X))
print(accuracy)
```

Naïve Bayes

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import cross_validation
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
from sklearn.cross_validation import train_test_split
```

```
from sklearn.utils.validation import check_array as check_arrays
from sklearn.metrics import classification_report
import os
#os.system("cls")

#Importando Dataset----------------------

df = pd.read_csv('muerte.txt')
X = np.array(df.drop(['14_Causa_CIE_a_Cuarto_digito'], 1))
y = np.array(df['14_Causa_CIE_a_Cuarto_digito'])

X_train, X_test, y_train, y_test = cross_validation.train_test_split(X, y,
test_size=0.3, random_state=0)

#Aplicando el Clasificador-----------------

nb = GaussianNB()

y_pred = nb.fit(X=X_train, y=y_train).predict(X_test);

print(nb.score(X,y))
score = nb.score(X_test, y_test)

target_names =
['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18',
'19','20','21','22','23','24','26','27','28','29','30','31','32','33','34','35','3
6','37','38','39','40','41','42','43','44','45','46','47','48','49','50','51','52'
,'53','54','55','56','57','58','59','60','61','62','63','64','65','66','67','68','
69','70','71','72','73','74','75','76','77','78','79','80','81','82','83','84','85
','86','87','88','89','90','91','92','93','94','95','96','97','98','99','100','101
','102','103','104','105','106','107','108','109','110','111','112','113','114','1
15','116','117','118','119','120','121','122','123','124','125','126','127','128',
'129','130','131','132','133','134','135','136','137','138','139','140','141','142
','143','144','145','146','147','148','149','150','151','152','153','154','155','1
56','157','158','159','160','161','162','163','164','165','166','167','168','169',
'171','172','173','174','175','176','177','178','179','180','181','182','183','184
','185','186','187','188','189','190','191','192','193','194','195','196','197','1
98','199','200','201','202','203','204','205','206','207','208','209','210','211',
'212','213','214','215','216','217','218','219','220','221','222','223','224','225
','226','227','228','229','230','231','232','233','234','235','236','237','238','2
39','240','241','242','243','244','245','24','247']

print(classification_report(y_test, y_pred, target_names=target_names))

#Imprime score y arreglo de predicción--------------------------

labels = nb.predict(X_test)
print labels
print 'Score test: '+str(score)

####################################
accuracy = nb.score(X, y)
nb.score(X_test, y_test)
print(nb.predict(X))
print(accuracy)
```

Naïve Bayes + PCA

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import cross_validation
from sklearn.decomposition import PCA
from sklearn.decomposition import RandomizedPCA
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
from sklearn.cross_validation import train_test_split
from sklearn.utils.validation import check_array as check_arrays
from sklearn.metrics import classification_report

#Importando Dataset----------------------

df = pd.read_csv('muerte.txt')
X = np.array(df.drop(['14_Causa_CIE_a_Cuarto_digito'], 1))
y = np.array(df['14_Causa_CIE_a_Cuarto_digito'])

X_train, X_test, y_train, y_test = cross_validation.train_test_split(X, y,
test_size=0.3, random_state=0)

#Aplicando PCA---------------------------

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
pca = PCA(n_components=2)
pca.fit(X_train)
X_t_train = pca.transform(X_train)
X_t_test = pca.transform(X_test)

pca = RandomizedPCA(n_components=2)
pca.fit_transform(X [y])
RandomizedPCA(copy=True, iterated_power=3, n_components=2,
 random_state=None, whiten=False)

print(pca.explained_variance_ratio_)

#Aplicando el Clasificador-----------------
nb = GaussianNB()
y_pred = nb.fit(X=X_train, y=y_train).predict(X_test);

-------------------------------------------

target_names =
['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18',
'19','20','21','22','23','24','26','27','28','29','30','31','32','33','34','35','3
6','37','38','39','40','41','42','43','44','45','46','47','48','49','50','51','52'
,'53','54','55','56','57','58','59','60','61','62','63','64','65','66','67','68','
69','70','71','72','73','74','75','76','77','78','79','80','81','82','83','84','85
','86','87','88','89','90','91','92','93','94','95','96','97','98','99','100','101
```

```
','102','103','104','105','106','107','108','109','110','111','112','113','114','1
15','116','117','118','119','120','121','122','123','124','125','126','127','128',
'129','130','131','132','133','134','135','136','137','138','139','140','141','142
','143','144','145','146','147','148','149','150','151','152','153','154','155','1
56','157','158','159','160','161','162','163','164','165','166','167','168','169',
'171','172','173','174','175','176','177','178','179','180','181','182','183','184
','185','186','187','188','189','190','191','192','193','194','195','196','197','1
98','199','200','201','202','203','204','205','206','207','208','209','210','211',
'212','213','214','215','216','217','218','219','220','221','222','223','224','225
','226','227','228','229','230','231','232','233','234','235','236','237','238','2
39','240','241','242','243','244','245','24','247']

print(classification_report(y_test, y_pred, target_names=target_names))

#Imprime score y arreglo de predicción---------------------------

labels = nb.predict(X_test)
print labels
print 'Score test: '+str(score)

labels = nb.predict(X_test)
print labels
score = nb.score(X_test, y_test)
print 'Score test: '+str(score)

###############
accuracy = nb.score(X, y)
nb.score(X_test, y_test)
print(nb.predict(X))
print(accuracy)
```

Regresión Logistica

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from datetime import datetime
from sklearn import linear_model
from sklearn import datasets
from sklearn.svm import l1_min_c
from sklearn.metrics import confusion_matrix
from sklearn import cross_validation
from sklearn.metrics import classification_report
from sklearn.cross_validation import train_test_split
import os
#os.system("cls")

#Importando Dataset----------------------

df = pd.read_csv('muerte.txt')
X = np.array(df.drop(['14_Causa_CIE_a_Cuarto_digito'], 1))
y = np.array(df['14_Causa_CIE_a_Cuarto_digito'])
```

```
X_train, X_test, y_train, y_test = cross_validation.train_test_split(X, y,
test_size=0.3, random_state=0)

#Aplicando el Clasificador-----------------
#visita la siguiente dirección web para saber los detalles de cada parametron.
# http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Logistic
#Regression.html

logreg = linear_model.LogisticRegression(C=0.01, class_weight=None, dual=False,
 fit_intercept=True, intercept_scaling=2, max_iter=248,
 multi_class='multinomial', n_jobs=1, penalty='l2', random_state=None,
 solver='lbfgs', tol=0.01, verbose=0, warm_start=False)

y_pred = logreg.fit(X_train, y_train).predict(X_test)
#Realizando el Reporte--------------------------

print y_pred[0:20]
print y_test[0:20]

target_names =
['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18',
'19','20','21','22','23','24','26','27','28','29','30','31','32','33','34','35','3
6','37','38','39','40','41','42','43','44','45','46','47','48','49','50','51','52'
,'53','54','55','56','57','58','59','60','61','62','63','64','65','66','67','68','
69','70','71','72','73','74','75','76','77','78','79','80','81','82','83','84','85
','86','87','88','89','90','91','92','93','94','95','96','97','98','99','100','101
','102','103','104','105','106','107','108','109','110','111','112','113','114','1
15','116','117','118','119','120','121','122','123','124','125','126','127','128',
'129','130','131','132','133','134','135','136','137','138','139','140','141','142
','143','144','145','146','147','148','149','150','151','152','153','154','155','1
56','157','158','159','160','161','162','163','164','165','166','167','168','169',
'171','172','173','174','175','176','177','178','179','180','181','182','183','184
','185','186','187','188','189','190','191','192','193','194','195','196','197','1
98','199','200','201','202','203','204','205','206','207','208','209','210','211',
'212','213','214','215','216','217','218','219','220','221','222','223','224','225
','226','227','228','229','230','231','232','233','234','235','236','237','238','2
39','240','241','242','243','244','245','24','247']

print(classification_report(y_test, y_pred, target_names=target_names))

logreg.fit(X,y)
print(logreg.score(X, y))

#Resultados prebios-------------------------------------
logreg.fit(X, y)
accuracy = logreg.score(X, y)
logreg.score(X, y)
logreg.score(X_test, y_test)

print(logreg.predict(X))
print(accuracy)
print(logreg.score(X,y))
print(logreg.score(X_test, y_test))
#######################
```

```
accuracy = logreg.score(X, y)
logreg.score(X_test, y_test)
print(logreg.predict(X))
print(accuracy)
```

Regresión Logistica + PCA

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import cross_validation
from sklearn.decomposition import PCA
from sklearn.decomposition import RandomizedPCA
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import linear_model, decomposition, datasets
from sklearn.metrics import confusion_matrix
from sklearn.cross_validation import train_test_split
from sklearn.utils.validation import check_array as check_arrays
from sklearn.pipeline import Pipeline
from sklearn.grid_search import GridSearchCV
from sklearn.cross_validation import KFold
from sklearn.metrics import classification_report
from sklearn.metrics import cohen_kappa_score
import os
#os.system("cls")

#Importando Dataset----------------------

df = pd.read_csv('muerte.txt')
X_digits = np.array(df.drop(['14_Causa_CIE_a_Cuarto_digito'], 1))
y_digits = np.array(df['14_Causa_CIE_a_Cuarto_digito'])

X_train, X_test, y_train, y_test = cross_validation.train_test_split(X_digits,
y_digits, test_size=0.3)

#Aplicando PCA---------------------------

pca = RandomizedPCA(n_components=2)
pca.fit(X_train)
X_t_train = pca.transform(X_train)
X_t_test = pca.transform(X_test)

pca.fit_transform(X_digits [y_digits])
RandomizedPCA(copy=True, iterated_power=3, n_components=2,
 random_state=None, whiten=False)

print(pca.explained_variance_ratio_)

#Aplicando el Clasificador-----------------

logistic = linear_model.LogisticRegression()
```

```
y_pred = logistic.fit(X_train, y_train).predict(X_test);

#Realizando el Reporte--------------------------------

print y_pred[0:20]
print y_test[0:20]

target_names =
['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18',
'19','20','21','22','23','24','26','27','28','29','30','31','32','33','34','35','3
6','37','38','39','40','41','42','43','44','45','46','47','48','49','50','51','52'
,'53','54','55','56','57','58','59','60','61','62','63','64','65','66','67','68','
69','70','71','72','73','74','75','76','77','78','79','80','81','82','83','84','85
','86','87','88','89','90','91','92','93','94','95','96','97','98','99','100','101
','102','103','104','105','106','107','108','109','110','111','112','113','114','1
15','116','117','118','119','120','121','122','123','124','125','126','127','128',
'129','130','131','132','133','134','135','136','137','138','139','140','141','142
','143','144','145','146','147','148','149','150','151','152','153','154','155','1
56','157','158','159','160','161','162','163','164','165','166','167','168','169',
'171','172','173','174','175','176','177','178','179','180','181','182','183','184
','185','186','187','188','189','190','191','192','193','194','195','196','197','1
98','199','200','201','202','203','204','205','206','207','208','209','210','211',
'212','213','214','215','216','217','218','219','220','221','222','223','224','225
','226','227','228','229','230','231','232','233','234','235','236','237','238','2
39','240','241','242','243','244','245','24','247']

print (classification_report(y_test, y_pred, target_names=target_names))

logistic.fit(X_digits, y_digits)
accuracy = logistic.score(X_digits, y_digits)
logistic.score(X_digits, y_digits)

print(accuracy)
print (logistic.score(X_digits, y_digits))

###################

accuracy = logistic.score(X_digits, y_digits)
logistic.score(X_test, y_test)
print(logistic.predict(X_digits))
print(accuracy)
```

SVM

```
import numpy as np
from sklearn import datasets
from sklearn import svm
from sklearn import cross_validation
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cross_validation import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.svm import SVC
```

```
from sklearn.metrics import classification_report
from sklearn import preprocessing, cross_validation, neighbors

#Importando Dataset----------------------

df = pd.read_csv('muerte.txt')
X = np.array(df.drop(['14_Causa_CIE_a_Cuarto_digito'], 1))
y = np.array(df['14_Causa_CIE_a_Cuarto_digito'])

X_train, X_test, y_train, y_test = cross_validation.train_test_split(X, y,
test_size=0.3, random_state=0)

#Aplicando el Clasificador-----------------
#Visita la siguiente dirección web para conocer los detalles de cada parámetro.
# http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

clf = svm.SVC(C=1.0, cache_size=1, class_weight=None, coef0=0.0,
 decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
 max_iter=-1, probability=False, random_state=None, shrinking=True,
 tol=0.001, verbose=False)

y_pred = clf.fit(X_train, y_train).predict(X_test)

clf.score(X_test, y_test)
print(clf.score(X,y))
print 'score = ', clf.score(X_test, y_test)

#Realizando el Reporte--------------------------

print y_pred[0:20]
print y_test[0:20]

target_names =
['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18',
'19','20','21','22','23','24','26','27','28','29','30','31','32','33','34','35','3
6','37','38','39','40','41','42','43','44','45','46','47','48','49','50','51','52'
,'53','54','55','56','57','58','59','60','61','62','63','64','65','66','67','68','
69','70','71','72','73','74','75','76','77','78','79','80','81','82','83','84','85
','86','87','88','89','90','91','92','93','94','95','96','97','98','99','100','101
','102','103','104','105','106','107','108','109','110','111','112','113','114','1
15','116','117','118','119','120','121','122','123','124','125','126','127','128',
'129','130','131','132','133','134','135','136','137','138','139','140','141','142
','143','144','145','146','147','148','149','150','151','152','153','154','155','1
56','157','158','159','160','161','162','163','164','165','166','167','168','169',
'171','172','173','174','175','176','177','178','179','180','181','182','183','184
','185','186','187','188','189','190','191','192','193','194','195','196','197','1
98','199','200','201','202','203','204','205','206','207','208','209','210','211',
'212','213','214','215','216','217','218','219','220','221','222','223','224','225
','226','227','228','229','230','231','232','233','234','235','236','237','238','2
39','240','241','242','243','244','245','24','247']

print (classification_report(y_test, y_pred, target_names=target_names))

accuracy = clf.score(X, y)
clf.score(X_test, y_test)
```

```
print(clf.predict(X))
print(accuracy)
```

SVM + PCA

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import preprocessing, cross_validation
import pandas as pd
from sklearn import datasets
from sklearn.cross_validation import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.decomposition import PCA
from sklearn.decomposition import RandomizedPCA
from sklearn.svm import SVC
from sklearn import cross_validation
from sklearn.metrics import classification_report
from sklearn import preprocessing, cross_validation, neighbors

#Importando Dataset----------------------

df = pd.read_csv('muerte.txt')
X = np.array(df.drop(['14_Causa_CIE_a_Cuarto_digito'], 1))
y = np.array(df['14_Causa_CIE_a_Cuarto_digito'])

X_train, X_test, y_train, y_test = cross_validation.train_test_split(X, y,
test_size=0.3, random_state=0)

#Aplicando PCA------------------

pca = PCA(n_components=2) # adjust yourself
pca.fit(X_train)
X_t_train = pca.transform(X_train)
X_t_test = pca.transform(X_test)

pca = RandomizedPCA(n_components=2)
pca.fit_transform(X [y])
RandomizedPCA(copy=True, iterated_power=3, n_components=2,
 random_state=None, whiten=False)

print(pca.explained_variance_ratio_)


#Aplicando el Clasificador------------------

classifier = SVC(C=1.0, cache_size=1, class_weight=None, coef0=0.0,
 decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
 max_iter=-1, probability=False, random_state=None, shrinking=True,
 tol=0.001, verbose=False)

y_pred = classifier.fit(X_train, y_train).predict(X_test)

#Realizando el Reporte-------------------------------
```

```
print y_pred[0:20]
print y_test[0:20]

target_names =
['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18',
'19','20','21','22','23','24','26','27','28','29','30','31','32','33','34','35','3
6','37','38','39','40','41','42','43','44','45','46','47','48','49','50','51','52'
,'53','54','55','56','57','58','59','60','61','62','63','64','65','66','67','68','
69','70','71','72','73','74','75','76','77','78','79','80','81','82','83','84','85
','86','87','88','89','90','91','92','93','94','95','96','97','98','99','100','101
','102','103','104','105','106','107','108','109','110','111','112','113','114','1
15','116','117','118','119','120','121','122','123','124','125','126','127','128',
'129','130','131','132','133','134','135','136','137','138','139','140','141','142
','143','144','145','146','147','148','149','150','151','152','153','154','155','1
56','157','158','159','160','161','162','163','164','165','166','167','168','169',
'171','172','173','174','175','176','177','178','179','180','181','182','183','184
','185','186','187','188','189','190','191','192','193','194','195','196','197','1
98','199','200','201','202','203','204','205','206','207','208','209','210','211',
'212','213','214','215','216','217','218','219','220','221','222','223','224','225
','226','227','228','229','230','231','232','233','234','235','236','237','238','2
39','240','241','242','243','244','245','24','247']

print (classification_report(y_test, y_pred, target_names=target_names))

accuracy = classifier.score(X, y)
classifier.score(X_test, y_test)
print(classifier.predict(X))
print(accuracy)
```

REFERENCIAS

Accuracy_score. (2017). *Accuracy classification score. Scikit-learn developers.* Recuperado de http://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

Agresti, A. (2002). *Categorical data analysis*. New York: John Wiley and Sons.

Alférez, G. H., Jiménez, J., Hernández-Navarro, H., González, M., Domínguez, R., Briones, A., y Hernández-Villalvazo, H. (2016a, noviembre). *Aplicación de ciencia de datos en las historias clínicas de una clínica ubicada en el noreste de México para corroborar la relación entre caries dental y diabetes*. Memorias del 2o Congreso de Investigación Universitaria de la División Interamericana (CIDIA 2016), Montemorelos, México.

Alférez, G. H., Jiménez, J., Hernández-Navarro, H., González, M., Domínguez, R., Briones, A. y Hernández-Villalvazo, H. (2016b, julio). *Application of data science to discover the relationship between dental caries and diabetes in dental records*. Memorias de la 2016 International Conference on Health Informatics and Medical Systems, Las Vegas, NV, USA.

Antti Ajanki A. (2007). *Example of k-nearest neighbour classificationnb*. Recuperado de https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#/media/File:KnnClassification.svg

Auditoría Superior de la Federación. (2013). *Evaluación de la política pública del primer nivel de la atención en salud*. Recuperado de http://www.asf.gob.mx/Trans/Informes/IR2013i/Documentos/Auditorias/2013_1208_a.pdf

Bahgat, E. M., Rady, S. y Gad, W. (2015, noviembre). An e-mail filtering approach using classification techniques. En T. Gaber, A. Hassanien, N. Bendary y N. Dei (Eds.), *1st International Conference on Advanced Intelligent System and Informatics.* Beni Suef, Egypt: Springer International Publishing.

Beleites, C., Neugebauer, U., Bocklitz, T., Krafft, C. y Popp, J. (2013). Sample size planning for classification models. *Analytica Chimica Acta, 760*, 25-33. doi:10.1016/j.aca.2012.11.007

Brownlee, J. (2014). *Classification accuracy is not enough: More performance measures you can use*. Recuperado de http://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/

Camargo-Vega, J. J., Camargo-Ortega, J. F. y Joyanes-Aguilar, L. (2015). Knowing the Big Data. *Revista Facultad de Ingeniería*, 24(38), 63-77.

Coordinación de Estrategia Digital Nacional. (2017). *México recibe premio por participación en Carta Internacional de Datos Abiertos*. Recuperado de https://datos.gob.mx/blog/mexico-recibe-premio-por-participacion-en-carta-internacional-de-datos-abiertos?category=noticias&tag=educacion

Córdova Villalobos, J. Á., Barriguete Meléndez, J. A., Lara Esqueda, A., Barquera, S., Rosas Peralta, M., Hernández Ávila, M., . . . Aguilar-Salinas, C. A. (2008). Las enfermedades crónicas no transmisibles en México: sinopsis epidemiológica y prevención integral. *Salud Pública de México, 50*(5), 419-427.

Davenport, D. J. y Patil, T. H. (2012). Data scientist: The sexiest job of the 21st Century. *Harvard Business Review*, *90*(10), 70-76

de Dios, J. G., Sempere, A. P. y Aleixandre-Benavent, R. (2007). Las publicaciones biomédicas en España a debate (II): las 'revoluciones' pendientes y su aplicación a las revistas neurológicas. *Neurol, 44*(2), 101-112. Recuperado de http://www.publicacions.ub.es/refs/Articles/pubiomedicas2.pdf

De la Fuente Fernández, S. (2011). *Regresión logística*. Recuperado de http://www.estadistica.net/ECONOMETRIA/CUALITATIVAS/LOGISTICA/regresion-logistica.pdf

Dhar, V. (2013). Data science and prediction. *Communications of the ACM, 56*(12), 64-73.

Dirección General de Información en Salud. (2016). *Datos abiertos – mortalidad materna.* Recuperado de http://www.dgis.salud.gob.mx/descargas/datosabiertos/mortalidad_materna_2002-2013.zip

Eng, N. (2016). *Making our moms proud: Reducing maternal mortality in Mexico*. Recuperado de https://dssg.uchicago.edu/2014/08/04/making-our-moms-proud-reducing-maternal-mortality-in-mexico/

f1_score. (2017). *Compute the F1 score, also known as balanced F-score or F-measure. Scikit-learn developers*. Recuperado de http://scikit-learn.org/stable/modules/ generated/sklearn.metrics.f1_score.html

Fan, S., Geissmann, Q., Lakatos, E., Lukauskas, S., Ale, A., Babtie, A. C., … Stumpf, M. P. H. (2016). MEANS: Python package for moment expansion approximation, inference and simulation. *Bioinformatics, 32*(18), 2863–2865. doi: org/10.1093/bioinformatics/btw229

Garg, R., Dong, S., Shah, S. y Jonnalagadda, S. R. (2016). *A bootstrap machine learning approach to identify rare disease patients from electronic health records*. Recuperado de https://arxiv.org/ftp/arxiv/papers/1609/1609.01586.pdf

Garrido, C. (2015). *Clasificación multi-etiqueta en entornos jerárquicos* (Tesis de maestría). Universidad Politécnica de Valencia, Valencia, España.

González, F. A. (2015). Machine learning models in rheumatology. *Revista Colombiana de Reumatología, 22*(2), 77-78.

Guerrero, A. L. (2016). *Utilizan Twitter para predecir epidemias como la influenza*. Recuperado de http://www.conacytprensa.mx/index.php/centros-conacyt/6467-utilizan-twitter- para-predecir-epidemias-como-la-influenza-nota

Hall, M. J., Levant, S. y DeFrances, C. J. (2013). Trends in inpatient hospital deaths: National hospital discharge survey*, 2000-2010*. US Department of Health and Human Services, Centers for Disease Control and Prevention, *National Center for Health Statistics,118*, 1-8. Recuperado de https://www.cdc.gov/nchs/data/ databriefs/ db118.pdf

Han, J., Pei, J. y Kamber, M. (2011). Getting to know your data. *Data Mining: Concepts and Techniques, 3*(744), 39-81.

Hede, K. (2013). Project data sphere to make cancer clinical trial data publicly available. *Journal of the National Cancer Institute, 105*(16), 1159-1160.

Hernández Orallo, J., Ramírez Quintana, M. J. y Ferri Ramírez, C. (2004*). Introducción a la minería de datos*. Madrid: Pearson Prentice Hall.

Hernández Pérez, A. y García Moreno, M. A. (2013). Datos abiertos y repositorios de datos: nuevo reto para los bibliotecarios. *El Profesional de la Información*, *22*(3), 259-263. doi:10.3145/epi.2013.may.10

Herrera Zurita, A. y Duran Cals, J. (2016). *Aprendizaje automático para la detección de ataques informáticos* (Tesis de maestría). Universidad Autónoma de Barcelona, Barcelona, España

Hinton, G. E. y Sejnowski, T. J. (1999). *Unsupervised learning: Foundations of neural computation.* Cambridge: Massachusetts Institute of Technology press.

Ho, P. G. P. (2009). Multivariate time series support vector machine for multispectral remote sensing image classifications. *InTech, 16,* 323-346. doi: 10.5772/8286

Honnungar, S., Mehra, S. y Joseph, S. (2016). *Diabetic retinopathy identification and severity classification*. Stanford, CA: Stanford University.

Irimia-Diéguez, A., Blanco-Oliver, A. y Oliver-Alfonso, M. D. (2016). Modelización de la autosuficiencia de las instituciones microfinancieras mediante regresión logística basada en análisis de componentes principales. *Journal of Economics, Finance and Administrative Science, 21*(40), 30-38.

Jain, V., Zhou, W. y Men, Y. (2013). *Machine learning classification of kidney and lung cancer types*. Stanford, CA: Stanford University.

LaFleur, M. T. y Vélez, J. (2014). *Determinantes de la salud materna e infantil y de los objetivos de desarrollo del milenio en Honduras*. Recuperado de http:// www.un.org/en/development/desa/policy/capacity/presentations/honduras/ Determinantes-de-MIyMM-en-Honduras.pdf

Layton, M. D., Campillo Carrete, B., Ablanedo Terrazas, I. y Sánchez Rodríguez, A. M. (2010). Reducing maternal mortality in Mexico: Building vertical alliances for change. En J. Gaventa y R. McGee (Eds.), *Citizen action and national policy reform. Department for International Development* (pp. 89-91). London: Zed Books.

Lazer, D., Kennedy, R., King, G. y Vespignani, A. (2014). The parable of Google flu: Traps in big data analysis. *Science, 343*(6176), 1203-1205. doi: 10.1126/science.1248506

Maillo, J., Ramírez, S., Triguero, I. y Herrera, F. (2016). kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. *Knowledge-Based Systems, 117*, 3–15. doi:10.1016/j.knosys.2016.06.012

Marín, C., Alférez, G. H., Córdova, J. y González, V. (2015, junio). *Detection of melanoma through image recognition and artificial neural networks.* Conferencia presentada en el World Congress on Medical Physics and Biomedical Engineering, Toronto, Canada. doi:10 .1007/978-3-319-19387-8_204

Meza, C. M. (2014). *Estudio comparativo de técnicas de selección de características para la clasificación de lesiones de mama en ultrasonografía* (Tesis de maestría). Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Campus Ciudad Victoria, Tamaulipas, México.

Mhenni, F., Nguyen, N. y Choley, J. Y. (2016). SafeSysE: A safety analysis integration in systems engineering approach. *Institute of Electrical and Electronics Engineers, Systems Journal, 99*, 1-12. doi: 0.1109/JSYST.2016.2547460

Michalski, R. S., Carbonell, J. G. y Mitchell, T. M. (1983). Machine learning: An artificial intelligence approach. *Springer, 20*(2), 111-161. doi:10.1007/978-3-662-12405-5

Moon, B. K. (2010). *Estrategia mundial de salud de las mujeres y los niños*. Recuperado de http://www.who.int/pmnch/activities/jointactionplan/201009_gswch_sp.pdf

Ng, A. Y. y Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems, 2,* 841-848.

Open data chárter. (2013). *Who we are*. Recuperado de http://opendatacharter.net/who-we-are/

Organización Mundial de la Salud. (2015). *Mortalidad materna.* Recuperado de http://www.who.int/mediacentre/factsheets/fs348/es/

Organización Panamericana de Salud. (2014). *Situación de salud en las Américas: indicadores básicos 2014.* Recuperado de http://www2.paho.org/hq/index.php?option=com_content&view=article&id=7170%3A2012-health-situation-americas-health-indicators-2014&catid=2394%3Aregional-health-observatory-reports&Itemid=2395&lang=es

Organización Panamericana de Salud. (2015). *Iniciativa de la OPS/OMS busca reducir las muertes maternas por hemorragias en países de las Américas*. Recuperado de http://www.paho.org/hq/index.php?option=com_content&view=article&id=10592%3Apahowho-initiative-seeks-to-reduce-maternal-deaths-from-hemorrhage&catid=740%3Anews-press-releases&Itemid=1926&lang=es

Organización para la Cooperación y el Desarrollo Económicos. (2016). *Estudios de la OCDE sobre los sistemas de salud: México 2016.* Paris: OECD. doi:10.1787/9789264265523-es

Potter, J. E. (2014). Utilización de los servicios de salud materna en el México rural. *Salud Pública de México, 30*(3), 387-402.

Precision-Recall. (2016). *Metric to evaluate classifier output quality.* Recuperado de http://scikit-learn.org/0.15/auto_examples/plot_ precision_recall.html

Quiroz Huerta, G., Tepetla, C. S., Cortés Salazar, C., Rojo Contreras, W. y Morales Andrade, E. (2015). Morbilidad materna extremadamente grave en el Centro de Especialidades Médicas del estado de Veracruz, 2012*. Revista Comisión Nacional de Arbitraje Médico, 20*(4) 160-173.

Ramírez, A. O. (2010). *Python como primer lenguaje de programación*. Recuperado de http://webcem01.cem.itesm.mx:8005/publicaciones/primer_lenguaje_30_ jun_2010 .pdf

Ramos, J. M., González-Alcaide, G. y Gutiérrez, F. (2016). Análisis bibliométrico de la producción científica española en enfermedades infecciosas y en microbiología. *Enfermedades Infecciosas y Microbiología Clínica, 34*(3), 166-176.

Ramos Guadarrama, J., Hernández Areu, O. N. y Bruzón Hernández, J. M. (2016). Ensayos de pérdidas en vacío y con carga en transformadores mediante la adquisición de datos. *Ingeniería Energética, 37*(1), 73-80.

Real Academia Española. (2014). *Diccionario de la lengua española* (23ª ed.). Madrid: Autor.

Riza, L. S. (2015). *Data science and big data processing in R: Representations and software* (Tesis doctoral). Universidad de Granada, Granada, España.

Rollins, J. (2015). *Foundational methodology for data science*. Recuperado de http:// www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=IMW14824USEN

Saltz, J., Shamshurin, I. y Crowston, K. (2017, enero). *Comparing data science project management methodologies via a controlled experiment*. Ponencia presentada en la 50th Hawaii International Conference on System Sciences, Manoa, Hawaii. doi:10.125/41273

Sankar, A. (2015). *Intuition of naive bayes.* Recuperado de http://analyticstraining.com/2015/intuition-of-naive-bayes/

Schoenherr, T. y Speier-Pero, C. (2015). Data science, predictive analytics, and big data in supply chain management: Current state and future potential. *Journal of Business Logistics, 36*(1), 120-132.

Scikit-Learn. (2016). *Model evaluation: Quantifying the quality of predictions*. Recuperado de http://scikit-learn.org/stable/modules/model_evaluation.html#accuracy-score

Secretaría de Salud. (2016). *Temas prioritarios en salud*. Recuperado de http://www.gob.mx/salud/acciones-y-programas/temas-prioritarios-en-salud

Segura, A. (2014). Prevención, iatrogenia y salud pública. *Gaceta Sanitaria, 28*(3), 181-182.

Shlens, J. (2014). *A tutorial on principal component analysis*. Recuperado de https://arxiv.org/pdf/1404.1100.pdf

Shukla, S., Gupta, D. L. y Prasad, B. R. (2016). Comparative study of recent trends on cancer disease prediction using data mining techniques. *International Journal of Database Theory and Application, 9*(9), 107-118. doi: 10.14257/ljdta.2016.9.9.10

Valdivia, R., Navarrete, M. y Aracena, D. (2014). Oportunidades en datos abiertos. *Ingeniare. Revista Chilena de Ingeniería, 22*(4), 458-459. doi:10.4067/S0718-33052014000400001

Van Rossum, G. (2009). *El tutorial de Python*. Recuperado de http://docs .python.org.ar/tutorial/pdfs/TutorialPython2.pdf

Vinent, C., Milián, Y. C., Estrada, V. D. A. y Bonet, A. R. (2016). Módulo de monitoreo de software para el Sistema Gestión de Recursos de Hardware y Software. *Serie Científica de la Universidad de las Ciencias Informáticas, 9*(10), 11-21.

Wang, Z., Cui, X., Gao, L., Yin, Q., Ke, L. y Zhang, S. (2016). A hybrid model of sentimental entity recognition on mobile social media. *EURASIP Journal on Wireless Communications and Networking, 2016*(1), 253. doi:10.1186/s13638-016-0745-7