Pontificia Universidad Católica de Chile
IIC2433 - Minería de Datos
Prof. Karim Pichara

Proyecto - Entrega Final

7 de diciembre de 2018

Luciano Davico - Gregory Schuit

I. Introducción y problema a tratar

Los movimientos telúricos o terremotos es algo propio de la naturaleza de nuestro planeta y, en muchas ocasiones, causan catástrofes en la corteza terrestre. Como ya se sabe, una de las consecuencias de estos acontecimientos puede ser la ocurrencia de un tsunami, el cual tiene un efecto aún más devastador para la naturaleza. a gente y los seres vivos en general.

En el presente informe se explicarán las actuales medidas sobre el problema, seguido de la propuesta detallada sobre la implementación de un *Random Forest* para clasificar los terremotos según tipo de tsunami que provocan. Todo esto en base a datos recolectados desde hace décadas.

II. Crítica a las soluciones actuales

Hoy en día, los científicos observan el tamaño y el tipo de terremoto que precede a un tsunami para predecir su tamaño, ya que es la primera información que se obtiene en estas situaciones. Además, esta observación se acompaña por sistemas de boyas que son instalados en el fondo del océano para detectar cambios de presión y temperatura enviando reportes en tiempo real a expertos para que ellos analicen la situación. Sin embargo, estos sistemas de boyas han sido criticados por su alta tasa de fracaso. Las boyas frecuentemente se degradan y dejan de funcionar en el ambiente submarino, y los costos de mantención son demasiado altos, por lo que las boyas no son remplazadas rápidamente (Larry West, 2018).

En el contexto de Machine Learning, los alcances que hoy en día se han dado sobre el tema se enfocan en predecir las características de tsunami mediante el análisis del monitoreo de las bayas en el océano. Un método, llamado Time Reverse Imaging, analiza datos sobre el impacto del tsunami en la costa para recrear los orígenes del tsunami, y así poder clasificar los tsunamis apenas nacen en un terremoto. Este algoritmo está aún en desarrollo y se espera que de frutos cerca del 2021 debido al periodo de recolección de datos (Victoria Woollaston, 2016)

III. Propuesta

El trabajo realizado en este informe se atiene a ser capaces de predecir con cierta precisión, mediante un modelo de clasificación de Machine Learning llamado *Random Forest*, el suceso de un tsunami, dado el acontecimiento de un terremoto de ciertas características. En otras palabras, se pretende establecer un modelo capaz de clasificar un movimiento telúrico como aquel capaz de provocar un tsunami con una cierta

probabilidad de certeza. Esto puede servir de apoyo a los servicios de alerta hidrográficos y planes de contingencia.

El modelo mencionado anteriormente será aplicado a una base de datos de entrenamiento preprocesada de terremotos (de grado 5.5 o más en la escala de Richter) ocurridos desde el año 1900 hasta la actualidad. Respecto al preprocesamiento, se incluirá un nuevo label de variables categóricas de tsunamis con instancias que indican si ocurrió o no el evento, dado el terremoto particular. Este label se extrae desde otra base de datos con tsunamis, en el mismo rango de años que en la base de datos de terremotos. Los datos de entrenamiento tendrán la utilidad de ajustar el grado de precisión del modelo para posteriormente utilizar datos de prueba, los cuales clasificarán los terremotos venideros de acuerdo a si provocarán un maremoto o no. Cabe destacar que la base de datos de los tsunamis contiene una columna que indica qué terremoto fue el causante del evento, y que mediante este dato haremos la fusión de las bases de datos.

IV. Supuestos y decisiones

Para llevar a cabo el proyecto, ocuparemos los datos sísmicos con las features de latitude, longitud, profundidad y magnitud, siendo todas variables numéricas continuas. Por otra parte, el label a ocupar será el tipo de tsunami provocado, siendo el tipo 0 la ausencia de tsunami y 1 cuando u tsunami ocurre/ocurrirá, por lo que solo hay clases binarias. Idealmente la base de datos podría tener más features para incorporar otros factores influyentes en la existencia de un tsunami y así entrenar de mejor manera el algoritmo de clasificación. Sin embargo, la búsqueda de nuevas dimensiones de los datos no fue satisfactoria, debido a que no hay muchos datos registrados que contengan información distinta a la que utilizaremos. Es importante destacar también que dada la disparidad de clases existente en la base de datos, se abordará este problema con técnicas utilizadas en Machine Learning y mediante librerías de Python especializadas como *sklearn*, todo esto será detallado en la sección V.

Dada la naturaleza de los datos y la cantidad de datos que pudimos extraer, consideramos que el modelo de *Random Forest* en variable continua clasificará los eventos sísmicos con una alta tasa de acierto (Breiman et al., 1984). Asimismo, para robustecer el análisis utilizaremos tres modelos de clasificación adicionales, los cuales serán *Regresión Logística*, *Support Vector Machine*, el cual llamaremos desde ahora *SVM*, y un modelo de redes neuronales llamado *Multilayer Perceptron*. Se utilizará una metodología similar para cada uno de estos, a modo de obtener resultados que sean comparables.

V. Metodología

Como se ha mencionado anteriormente, el trabajo de preprocesamiento de la base de datos y de clasificación de clases se realizó completamente en Python. En primer lugar, se analizó la distribución de la variable Magnitud en la base de datos de terremotos, en la cual se estimó que si ocurría un terremoto de grado menor a 6.3, había una alta probabilidad de que no ocurra un tsunami. Esto es fácil de ver en la Figura 1, donde se detalla la distribución de la magnitud de los terremotos ocurridos y que producieron tsunamis:

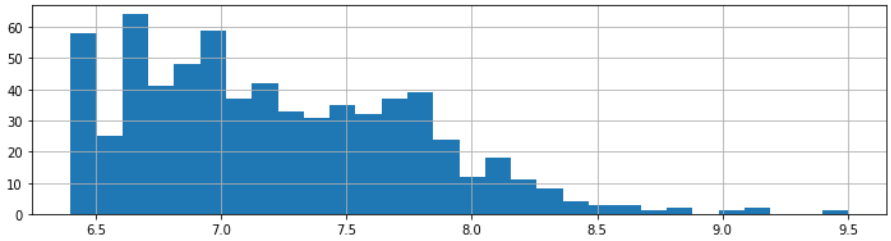

Figura 1: Tsunami's Magnitude Distribution.

Por otra parte, dadas las features de Latitud y Longitud, se presenta una figura que muestra los terremotos ocurridos en amarillo. Aquellos puntos de color morado indican que ocurrió un tsunami en aquella zona, dado el terremoto. Se puede observar que los puntos graficados forman el contorno de los continentes, lo que tiene sentido ya que en aquellos lugares hay mayor probabilidad de ocurrencia de terremotos, debido a que son zonas donde se topan distintas placas tectónicas. A continuación se muestra lo detallado:

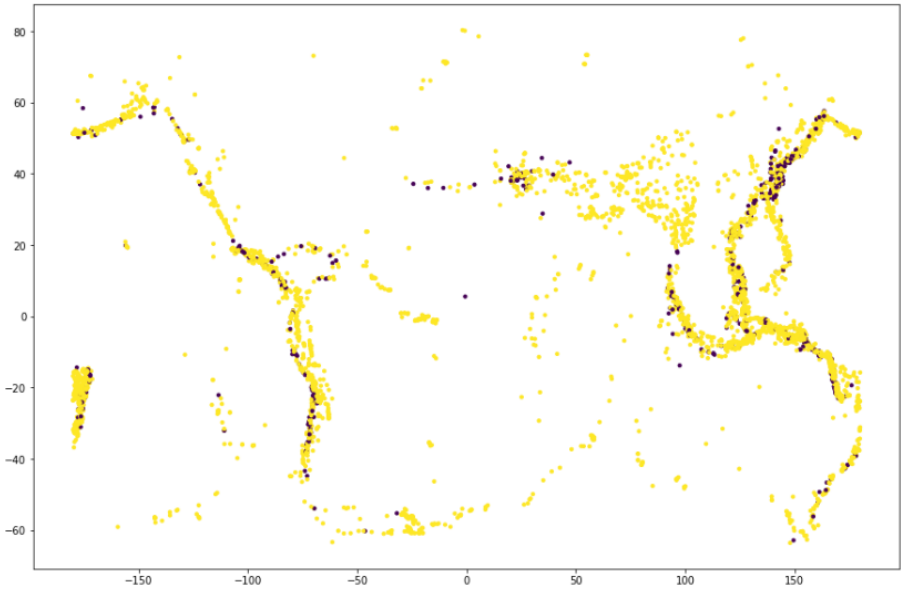

Figura 2: Earthquake's Map Distribution.

Podemos ver en el scatter plot, que si pintamos los terremotos que no causan tsunami de amarillo, y los que sí causan tsunami de morado, se trazan las zonas más sísmicas del planeta. Los puntos amarillos logran mostrar, a la izquierda de la visualización, la frontera oeste del continente americano, mientras que a la derecha podemos distinguir la frontera este de Asia y la polinesia. Por conocimiento general, esperábamos esto ya que estas zonas son el límite de la placa tectónica del pacífico, zona donde los volcanes y sismos abundan.

Basándonos en la información del gráfico en la Figura 1, se aplicó un filtro a la base de datos considerando únicamente terremotos de grado 6.3 o mayor. Esto implicó una pérdida importante de datos, ya que inicialmente contábamos con 415.677 observaciones y luego del filtro la cantidad de observaciones disminuyó a 4.930. A pesar de lo anterior, el filtro se mantuvo para la base de datos. En segundo lugar, dado que la base de datos de tsunamis y terremotos tienen las mismas features de Latitud, Longitud, Magnitud y Fecha,

procedimos a agregar el label 'tsunami' a la base de datos de terremotos buscando que los valores de estas features sean similares entre ambas bases de datos, considerando ciertos márgenes de tolerancia.

Respecto al trabajo realizado posteriormente, dado que en la base de datos habían dos clases notoriamente desbalanceadas, se decidió utilizar la técnica de *Up-Sample Minority Class* en el que se procesa el set de entrenamiento de forma que la base de datos contenga cantidades semejantes de datos de ambas clases. En particular, en la base de datos hay 4.259 clases de tipo 0 (no ocurre tsunami) y 671 clases de tipo 1 (ocurre tsunami). Con esta técnica aumenta la cantidad de predicciones correctas de la clase minoritaria, sin embargo, puede existir un impacto negativo en la precisión de la predicción. Al momento de definir el set de entrenamiento y set de test lo hicimos en la proporción 70:30, por lo que las magnitudes de cada set fueron 3.451 y 1.479, respectivamente. Debido a lo anterior, al aplicar *Up-Sample* al set de entrenamiento, este quedó con 5.962 observaciones.

Una vez con ambos set de datos definidos, se utilizaron los cuatro modelos de clasificación nombrados anteriormente y se calcularon medidas de evaluación de sus predicciones. Asimismo, se utilizó una curva ROC para observar de forma más didáctica y gráfica el desempeño de cada clasificador. Esta es una herramienta que evalúa la sensibilidad y especificidad de un modelo de clasificación binaria, de forma que se puede saber cuán cerca está cada modelo de predecir el valor real de un dato (del Valle Ana Rocío, 2017).

Cabe destacar que, dadas las características de este problema en particular, se debe poner el énfasis en obtener un alto número de tsunamis correctamente predichos, es decir, maximizar la cantidad de Verdaderos Positivos. Esto se evalúa mediante la medida Recall, la cual cuantifica la probabilidad de predecir la ocurrencia de un tsunami, dado que ha ocurrido uno. Esto es importante, ya que el costo de tener una predicción errónea de un tsunami puede traducirse en la pérdida de muchas vidas, dado que no se han ejecutado planes de evacuación en un tiempo suficientemente prudente.

VI. Resultados y Análisis

De acuerdo a los modelos utilizados, se evaluó cada uno de ellos utilizando una partición estratificada de los datos. Además, cada modelo se entrenó utilizando datos normales y datos a los que se les aplicó Up Sample. Los resultados y eficacia de estos modelos se reflejan en los valores de las métricas utilizadas, las cuales fueron Recall, Precision, F1-Score y Accuracy. Además se utilizó una Matriz de Confusión para mostrar la frecuencia de Los Verdaderos Positivos, Verdaderos Negativos, Falsos Positivos y Falsos Negativos.

El primer modelo que se evaluó fue *Random Forest*. En este modelo se generaron 75 árboles de decisión y una profundidad máxima de 10 ramas. Además, se consideró el parámetro *class weight='balanced'* para abordar la base de datos desbalanceada, los resultados a continuación:

Métricas de Random Forest sin Up-Sample

	precision	recall	f1-score	support
no provoca tsunami	0.93	0.93	0.93	1278
sí provoca tsunami	0.55	0.58	0.57	201
avg / total	0.88	0.88	0.88	1479

Accuracy: 0.8796484110885734

Matriz de Confusión:

```
[[1184   94]
 [  84  117]]
```

Figura 3: Random Forest's Metrics.

El segundo modelo evaluado fue *Logistic Regression*. En este modelo también se utilizó el parámetro *class weight='balanced'* para los mismo efectos del modelo anterior, los resultados a continuación:

Métricas de Regresión Logística sin Up-Sample

	precision	recall	f1-score	support
no provoca tsunami	0.94	0.78	0.85	1278
sí provoca tsunami	0.33	0.70	0.45	201
avg / total	0.86	0.77	0.80	1479

Accuracy: 0.7660581473968898

Matriz de Confusión:

```
[[993 285]
 [ 61 140]]
```

Figura 4: Logistic Regression's Metrics.

El tercer modelo evaluado fue *Support Vector Machine*. Este clasificador trata de encontrar un hiperplano divisor de los datos. Como podemos intuir de la naturaleza de los datos, estos no son separables por un hiperplano, por lo que es esperable que este clasificador tenga un mal rendimiento. Para este clasificador también se utilizó el parámetro *class weight='balanced'* y se utilizó un kernel sigmoide. A continuación sus métricas:

Métricas de Support Vector Machine sin Up-Sample

	precision	recall	f1-score	support
no provoca tsunami	0.86	0.51	0.64	1278
sí provoca tsunami	0.13	0.46	0.20	201
avg / total	0.76	0.50	0.58	1479

Accuracy: 0.49966193373901285

Matriz de Confusión:

```
[[647 631]
 [109  92]]
```

Figura 5: Support Vector Machine's Metrics.

El cuarto y último modelo evaluado fue el clasificador *Neural Network*. En particular, se utilizó la variante *Multilayer Perceptron*. Para este modelo se ocupó la función logística como función de activación y el algoritmo *lbfgs* para resolver el problema de optimización de determinación de los pesos, este algoritmo tiene un funcionamiento aproximadamente parecido al algoritmo de Newton. A continuación los resultados de las métricas:

Métricas de Multilayer Perceptron sin Up-Sample

	precision	recall	f1-score	support
no provoca tsunami	0.88	0.97	0.92	1278
sí provoca tsunami	0.42	0.14	0.21	201
avg / total	0.82	0.86	0.82	1479

Accuracy: 0.8573360378634213

Matriz de Confusión:

```
[[1240   38]
 [ 173   28]]
```

Figura 6: Multilayer Perceptron's Metrics.

Como se puede ver, el clasificador *Random Forest* es el que ha obtenido los mejores resultados de las métricas. Esto se puede ver ya que el Recall, Precision y Accuracy del algoritmo son los más altos, con una puntuación promedio de 0.88, 0.88 y 0.879, respectivamente. Sin embargo, estas medidas no reflejan el desempeño del algoritmo en la práctica, ya que la Precision y Recall del modelo para predecir un tsunami son relativamente bajas para las expectativas del clasificador. Esto es explicado por la baja frecuencia de la clase '1' en la base de datos, por lo que para el modelo no predecir estos datos no implica un impacto muy alto en las puntuaciones. De hecho, si el modelo predijera solo la clase mayoritaria seguiría teniendo buenos resultados de Precision y Accuracy. Pero no de Recall, ya que este mide la probabilidad de predecir los Verdaderos

Positivos (predecir un tsunami dado que efectivamente ocurrió).

Luego de este análisis, procedimos a trabajar con *Random Forest* y Regresion Logística para evaluar su rendimiento realizando algunas mejoras que se muestran a continuación.

Up-Sample

El método de Up-Sample, consiste en ponderar los datos de la clase minoritaria de tal manera que queden igual en frecuencia que los de la clase predominante. Esto resulta en una mayor influencia para los datos minoritarios, ya que el algoritmo los ve más veces y por lo tanto los considera más relevantes.

Gracias a este sampleo, logramos nivelar los scores de Random Forest y de la Regresion logística:

- Random Forest:
  - Precision: 0.48
  - Recall: 0.67
  - F-score: 0.56
  - Accuracy: 0.86

- Regresion Logistica:
  - Precision: 0.33
  - Recall: 0.70
  - F-score: 0.45
  - Accuracy: 0.76

División de continentes

Dada la característica de los datos de ser poco separables, llevamos a cabo una separación de los datos en 3 segmentos: América, Europa y África, y Asia. Esto fue a modo de simplificar los datos que se le entregaban a los modelos en cuanto a longitud y latitud, ya que si miramos los datos por segmentos, estos se ven más separables que todos juntos. Los resultados fueron los siguientes:

Random Forest:

- América:
  - Precision: 0.61
  - Recall: 0.17
  - F-score: 0.27
  - Accuracy: 0.87

- Centro:
  - Precision: 0.65
  - Recall: 0.07
  - F-score: 0.13
  - Accuracy: 0.87

- Asia:
  - Precision: 0.57
  - Recall: 0.46
  - F-score: 0.51
  - Accuracy: 0.88

Regresión Logística:

- América:
  - Precision: 0.16
  - Recall: 0.88
  - F-score: 0.27
  - Accuracy: 0.35

- Centro:
  - Precision: 0.13
  - Recall: 0.28
  - F-score: 0.18
  - Accuracy: 0.66

- Asia:
  - Precision: 0.16
  - Recall: 0.92
  - F-score: 0.28
  - Accuracy: 0.34

Según el F-Score, podemos ver que en América funcionan ambos algoritmos de la misma forma, pero si vemos la zona del centro, funciona mejor la regresión logística, mientras que en Asia funciona mejor el Random

Forest. Esto podría servir para unir modelos y lograr un resultado general mejor. Sin embargo, vemos que el análisis con todos los datos en conjunto logra un F-Score de 0.56 con random forest, que es mejor que cualquiera de las particiones creadas.

El análisis realizado anteriormente se puede obtener de una forma más didáctica y gráfica mediante una curva ROC. Esta indica la relación entre falsos positivos y verdaderos positivos en un algoritmo de clasificación binaria. Es decir, nos muestra qué tan bien predice cada uno de los clasificadores, de acuerdo a la sensibilidad (Recall) y la especificidad (1 - Precision) del algoritmo. El área que está debajo de la curva de cada algoritmo se llama AUC. Esta área se puede interpretar como la probabilidad de que un modelo clasifique un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio. A continuación se detalla la curva:

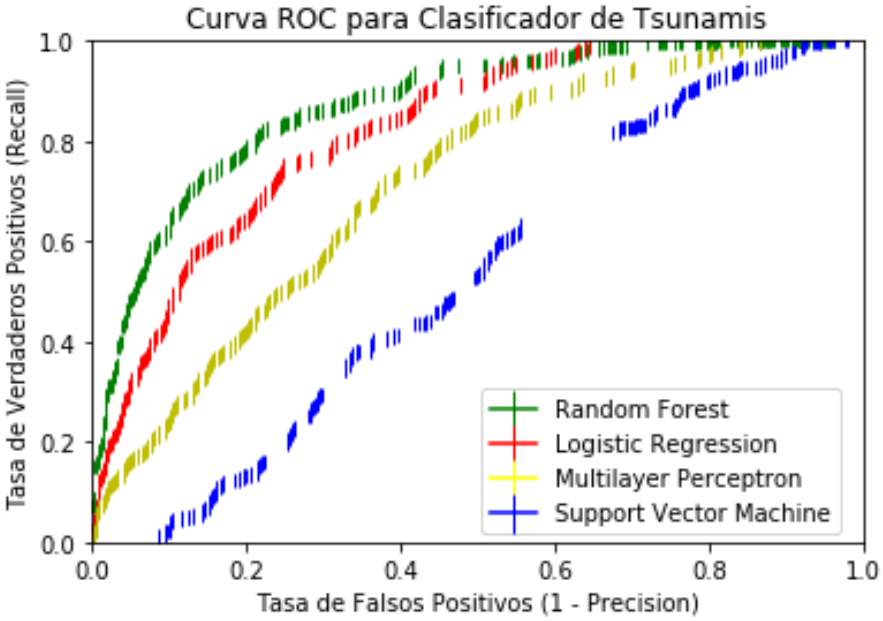

Figura 7: Receiver Operating Characteristic Curve

Se observa que el AUC del modelo Random Forest es el mayor de todos los clasificadores. Esta medida se utiliza en variados casos como un evaluador global de un modelo de clasificación.

VII. Base de Datos y Código Python

Para facilitar la coordinación y programación en equipo, decidimos utilizar la plataforma de GitHub, en el siguiente repositorio:

- Inicio del preprocesamiento: `https://github.com/gregschuit/Proyecto_IIC2433#proyecto_iic2433`

Para poder compartir la base de datos fácilmente, creamos una carpeta en Google Drive. Es importante destacar que el acceso debe ser mediante una cuenta uc.

- tsunamis.tsv: `https://drive.google.com/file/d/1ruavPv4NkhRvHoY5GIlEwPPKtH-3Lupu/view?usp=sharing`
- terremotos.csv: `https://drive.google.com/a/uc.cl/file/d/1jpTz2y5RBR0aFkEaECD3DAP3uIN2HSF6/view?usp=sharing`

VIII. Conclusión: dificultades y aprendizaje

En el presente proyecto se ha abordado un problema de clasificación binario con clases desbalanceadas. Se decidió utilizar cuatro modelos y así evaluar sus resultados mediante métricas. Los resultados de esto fueron concluyentes y categóricos, ya que el clasificador *Random Forest* fue el modelo más efectivo en todos los casos. En este sentido, se pudo apreciar las medidas de Recall y Precision fueron las más altas, por lo que al momento de predecir la ocurrencia de un tsunami este clasificador se torna como el más confiable.

Por otra parte, un aprendizaje importante fue que al momento de evaluar un clasificador en cuanto a rendimiento se debe considerar el contexto del problema. En este caso, el objetivo era predecir correctamente la ocurrencia de un tsunami y, por lo tanto, predecir la clase minoritaria de la base de datos. En este sentido, en la práctica se producen dos posibles caminos. Por un lado, un clasificador puede obtener una alta Precisión y un bajo Recall. Esto se traduce en que el modelo predice con alta precisión la clase mayoritaria. Esto es algo simple para el algoritmo, ya que en el proceso de aprendizaje la probabilidad de predecir la clase mayoritaria es alta. En la práctica esto indica que es fácil predecir que no ocurrirá un tsunami con certeza, lo que no significa que el modelo tendrá una buena predicción de cuando ocurrirá un tsunami. Para obtener una buena predicción de un tsunami que sí ocurrirá se requiere un Recall alto, en desmedro de una baja Precisión. Esto se refleja en la práctica en que un modelo predice muy bien cuando ocurrirá un tsunami, sin embargo, al tener una baja precisión existirán predicciones erróneas en cuanto a la no ocurrencia de un tsunami, ya que se predecirán más tsunamis de los que efectivamente ocurren. Esto puede tener costos en planes de contingencia y evacuación si el modelo es aplicado en un servicio real, sin embargo, puede ser un aporte para evitar pérdidas de vida, ya que es poco probable que el modelo no prediga un tsunami que sí ocurrirá.

Con respecto a las dificultades del proyecto, lo principal fue el desbalance de los datos, y la debilidad de las features. El primer obstáculo fue encontrar un punto óptimo de corte en la magnitud para filtrar los sísmos. Nuestra primera opción fue cortar en 5.5, para no eliminar tantos datos. Pero luego, analizando los modelos, preferimos cortar en 6.3 porque es el punto donde el desbalance es menor, y se obtuvo un mejor rendimiento en este punto.

En segundo lugar, realizamos una larga investigación para encontrar como mejorar el rendimiento del clasificador sin encontrar nada que se ajustara a los datos. Luego de decidirnos por escoger up-sample, logramos ver pequeños cambios en como random forest y regresion logística. Luego de estos resultados, pensamos en la siguiente técnica de separar por continentes, pero los resultados tampoco fueron favorables.

Finalmente, nos dimos cuenta de que el gran problema luego de balancear los datos filtrando por magnitud, era la calidad de las features. Es por esto que, como a trabajo a futuro, nuestras propuestas se basarían en darle más información al algoritmo incluyendo más columnas en la base de datos, tales como si el terremoto fue en mar o fue terrestre, duración del terremoto, distancia a la costa, información sobre las ondas del terromotos, etc.

IX. Referencias

Proyecto antiguo

- Radio Cooperativa. (2015). Buses del Transantiago circulan más lento que hace tres años. Lugar de publicación: `https://www.cooperativa.cl`. Recuperado de: `https://www.cooperativa.cl/noticias/pais/transportestransantiago/buses-del-transantiago-circulan-mas-lento-que-hace-tres-anos/2015-02-04/151139.html`

- Ayobami Ephraim Adewale. (2017). Deep Learning Based Origin Destination Matrix Prediction Through GPS Data. Recuperado de: http://ds.cs.ut.ee/courses/course-files/Ayobami-project.pdf/view
- Nicholas G. Polson, Vadim O. Sokolov. (2017). Deep Learning for Short-Term Traffic Flow Prediction. University of Chicago.
- Michael A. Nielsen. (2015). Neural Networks and Deep Learning. Toronto, Canada: Determination Press. Recuperado de: http://neuralnetworksanddeeplearning.com/chap2.html

Referencias nuevas

- USGS Earthquake Hazards Program, earthquake.usgs.gov, URL: https://earthquake.usgs.gov/earthquakes/
- National center for environmental information: National oceanic and atmospheric administration. https://www.ngdc.noaa.gov
- Victoria Wollaston, 2016, Revista Wired,
URL: https://www.wired.co.uk/article/the-algorithm-that-can-predict-when-a-tsunami-will-strike
- Larry West, 2018 ThoughtCo.,
URL: https://www.thoughtco.com/tsunami-detection-and-warning-1203697
- Leo Breiman, J. H Friedman, R. A Olshen, and C. J. Stone. Classification and regression trees. Wadsworth Brooks/Cole Advanced Books Software, 1984.
- Ana Rocío del Valle Benavides. (2017). Curvas ROC (Receiver-Operating-Characteristic) y sus aplicaciones. Universidad de Sevilla. URL: https://idus.us.es/xmlui/bitstream/handle/11441/63201/Valle%20Benavides%20Ana%20Roc%C3%ADo%20del%20TFG.pdf?sequence=1