Documentación codificación automática

Presentación

Tradicionalmente, los procesamientos estadísticos en el INE han recurrido a una estrategia de codificación manual para las variables de respuesta abierta. De este modo, personas entrenadas en el uso de clasificadores leen cada uno de los registros y asignan el código más adecuado, sobre la base de criterios predefinidos. Este procedimiento, por ser intensivo en trabajo manual, requiere una gran cantidad de tiempo y presenta algunas limitaciones para lograr una completa uniformidad en la utilización de criterios.

Con el objeto de hacer más eficiente el uso de recursos y mejorar la calidad de los datos, durante los últimos años la institución ha avanzado en estrategias automatizadas de codificación, basadas en técnicas de aprendizaje de máquinas (*machine learning*). El conocimiento acumulado y el desarrollo de competencias han permitido que estas técnicas sean aplicadas en la Encuesta Nacional de Empleo (ENE) y en la Prueba Piloto EPF 2021. En el caso de la ENE, desde 2020 [CHEQUEAR] se está implementando una metodología similar a la desarrollada por Guerrero y Cabezas (2019), quienes muestran resultados para CAENES y CIUO [1]. En el caso de la Prueba Piloto EPF, se utilizó una técnica de machine learning para codificar los datos de gasto y ocupación para CCIF y CIUO-08.

Sin duda, la utilización de este tipo de herramientas supone un avance en la elaboración de estadísticas, ya que ello permite disminuir tiempos de procesamiento y mejorar la calidad, sin embargo, los métodos de clasificación basados en *machine learning* presentan algunas limitaciones, que no deben perderse de vista. Una de las más importantes guarda relación con la posible desactualización de los datos que se utilizan para el entrenamiento de los modelos.

Cuando los datos de entrenamiento comienzan a perder representatividad o, dicho de otro modo, cuando las características del conjunto de entrenamiento empiezan a diferir de manera importante de las características de los datos para los cuales se busca hacer una predicción, lo más probable es que el rendimiento real sea más bajo que el estimado durante el proceso de entrenamiento. En la medida en que los nuevos datos se van distanciando del conjunto inicial, lo esperable es que las predicciones se vayan deteriorando progresivamente, afectando de manera importante la calidad estadística de los productos que usen esta estrategia.

La desactualización puede provenir de varias fuentes. Una de las más importantes se relaciona con cambios metodológicos, tales como el tránsito de encuestas en papel a dispositivos electrónicos, modificaciones en los cuestionarios, incorporación de nuevas instrucciones en los operativos de campo, entre otras. Una segunda fuente de desactualización es el dinamismo propio de los fenómenos que las encuestas de hogares miden. A modo de ejemplo, tanto la ocupación como el consumo de los hogares constituyen fenómenos que van cambiando en el tiempo. Así, al alero de cambios tecnológicos, van surgiendo nuevas ocupaciones, a la vez que otras desaparecen. En el caso del consumo de los hogares ocurre algo similar, ya que el mercado constantemente va cambiando la oferta de bienes y servicios, al mismo tiempo que se modifican las preferencias de los hogares.

Todos estos cambios añaden dificultades a las estrategias de codificación basadas en *machine learning* e imponen la necesidad de monitorear constantemente los modelos que están en producción, con el objeto de introducir ajustes, en caso de que ello sea necesario. En ese sentido, el presente proyecto tiene como objetivo principal avanzar en la actualización de datos de entrenamiento para los clasificadores de CAENES y CIUO. Dado que la institución se encuentra en un proceso de migración hacia dispositivos electrónicos de captura, es deseable contar con datos de entrenamiento que provengan de levantamientos con dispositivos de este tipo. El motivo de ello es que muy probablemente el modo en el que se registra la información en un dispositivo electrónico difiera

del modo en el que las personas lo harían en un formato basado en papel. Es así que el presente proyecto pone a disposición dos sets de datos (CAENES y CIUO) provenientes de levantamientos CAPI (*Computer assisted personal interview*). En la elaboración de estos sets de entrenamiento, se ha intentado utilizar altos estánderes de exigencia, con el objeto de ofrecer datos con una alta calidad, que permitan llevar a cabo próximos entrenamientos.

Como objetivo secundario, este proyecto intenta subsanar la aun escasa diseminación en lo que respecta al desarrollo y uso de técnicas basadas en *machine learning* dentro de la institución. Si bien se han llevado a cabo avances importantes en esta materia, aún existe una brecha de competencias que es posible estrechar. Para ello, se disponibilizan las rutinas utilizadas y al mismo tiempo se pone en funcionamiento una herramienta que permite utilizar los modelos entrenados de manera relativamente sencilla. Con esto último se busca que los esfuerzos realizados por este equipo puedan ser de utilidad para operaciones que tengan necesidades similares de codificación, constribuyendo con esto a un uso más eficiente de los recursos públicos.

Machine learning para la codificación de CAENES y CIUO

Codificación automatizada de textos

Las encuestas de hogares suelen contener preguntas abiertas en formato de texto libre. Esta información puede tener relación con el mercado del trabajo, el consumo de los hogares, la victimización, entre otros. Para que dicha información sea utilizada con fines estadísticos se debe recurrir a algún sistema de codificación estandarizado, de manera tal de que cada texto sea asociado a un número dentro de un sistema de clasificación. Históricamente, este trabajo ha sido realizado por codificadores entrenados para esta tarea. Así, personas que conocen en detalle los clasificadores leen cada uno de los textos y asignan un código conforme a criterios pre establecidos. Esta estrategia, por tanto, es intensiva en trabajo manual.

Es posible identificar, al menos, 2 debilidades de una estrategia basada en codificación manual:

- Requiere una gran cantidad de tiempo, ya que es necesario que cada uno de los textos sea leído y luego codificado por una persona. Típicamente, una muestra tiene varios miles de registros, lo que implica que el trabajo manual destinado a esta tarea sea significativo, generando tiempos de espera y costos en horas persona.
- El hecho de que la asignación de códigos descanse en las decisiones de distintas personas, hace bastante complejo el aseguramiento de una completa uniformidad en la aplicación de criterios. Si bien cada clasificador cuenta con un marco conceptual definido, la aplicación de criterios en la práctica no siempre resulta una tarea sencilla. La información levantada en terreno muchas veces puede ser insuficiente o ambigua, lo cual hace que se torne difícil la utilización de criterios rígidos.

A raíz de las dos situaciones mencionadas, es que se torna deseable la utilización de estrategias automáticas de codificación. Un proceso automático aplicado a la codificación permite disminuir significativamente las horas empleadas para dicha tarea y, al mismo tiempo, elimina el espacio para la discrecionalidad al momento de asignar un código.

En lo que respecta a la automización de la codificación en producción de estadísticas oficiales, es posible identificar dos enfoques principales: codificación basada en reglas y codificación basada en métodos de aprendizaje de máquinas (*machine learning*). En el primero de estos enfoques, el analista genera reglas a priori, mediante las cuales un programa o rutina asigna un código. La mayor fortaleza de este enfoque radica en que no es necesario contar con datos históricos, pues solo se requiere que el analista identifique las reglas y las

programe en una rutina. Una vez realizada la tarea de identificación de reglas, los datos pasan a través del programa y el código es asignado. La desventaja más importante y evidente es que se hace necesario que alguien escriba las reglas, lo cual constituye una tarea ardua, ya que por lo general el número de condiciones que se deben escribir no es despreciable. De hecho, es muy poco probable que se llegue a codificar el 100% de los registros mediante esta estrategia.

El segundo enfoque está basado en el aprendizaje que un algoritmo puede obtener de los datos que se han etiquetado en el pasado. A diferencia de la estrategia anterior el analista no crea las reglas, sino que estas son aprendidas por un algoritmo, el cual, después de un proceso de entrenamiento, es capaz de codificar registros nuevos gracias al aprendizaje previo. Naturalmente, esta estrategia supone la existencia de datos codificados con anterioridad, lo cual no siempre ocurre o no ocurre de manera óptima. Por ejemplo, es posible que la cantidad de datos codificados sea muy pequeña, que estos no sean representativos del fenómeno en cuestión o que, simplemente, la calidad del etiquetado no sea la más adecuada. En cada uno de estos casos se ve afectada la posibilidad de que un algoritmo "aprenda" correctamente. Ahora bien, cuando la fuente de datos de entrenamiento sí cumple ciertos requisitos de calidad, se puede llevar a cabo un proceso de aprendizaje, mediante el cual es posible codificar registros nuevos en poco tiempo y a un costo muy bajo.

Dependiendo del contexto y de los datos históricos con los que se cuente, un equipo debería privilegiar una estrategia u otra. A priori, no es tan sencillo determinar qué estrategia es la más eficiente, ya que existen consideraciones de costos que deben ser sopesadas en el marco de cada proceso productivo. Para operaciones estadísticas coyunturales, cuya metodología no experimenta cambios importantes, puede ser de gran utilidad un enfoque basado en *machine learning*, ya que un mismo entrenamiento puede ser utilizado para muchos levantamientos. Por otro lado, operaciones estadísticas de naturaleza estructural, posiblemente, no se vean demasiado benefiadas por una estrategia de *machine learning*, ya que en periodos extensos de tiempo es muy problable que los clasificadores experimenten cambios relevantes, lo cual dejaría en obsolecencia los datos de entrenamiento.

El INE cuenta con experiencia institucional en el uso de ambos enfoques sin embargo, en lo que respecta al presente documento, se ahondará solamente en la estrategia de *machine learning*.

Limitaciones de un método de machine learning

Una codificación automática basada en machine learning tiene una serie de ventajas, sin embargo, también presenta algunas limitaciones que es preciso tener en consideración al momento de poner en producción esta estrategia. El supuesto más importante en una estrategia basada en machine learning es que los datos de entrenamiento son representativos de los datos nuevos que se pretende codificar. Si dicho supuesto no se cumple, la estimación del error de predicción estará sesgada y, muy probablemente, el error real sea mayor al estimado. En ese sentido, nunca debe perderse de vista la relación que existe entre los datos de entrenamiento y aquellos para los que se está haciendo una predicción. En el contexto de las estadísticas oficiales, este problema podría originarse por varias causas. Algunas de las más importantes son:

- El dinamismo de los fenómenos en cuestión: la ocupación, el consumo o la producción son fenómenos que van cambiando constantemente. En el caso del mercado del trabajo, es posible que surjan nuevas ocupaciones que no estén consideradas dentro del conjunto de datos de entrenamiento. Un ejemplo de ello son las ocupaciones que surgen al alero de cambios técnológicos [2]. En el caso del consumo de los hogares ocurre algo similar, ya que constantemente nuevos bienes y servicios son lanzados al mercado, lo que introduce presión sobre el proceso de codificación automática.
- Cambios metodológicos durante el levantamiento: Podría ocurrir que el modo en el que se capturan los datos introduzca que sus características cambien. Algunos ejemplos de ello son: cambios en el freseo de las preguntas, migración de papel a dispositivos electrónicos, cambio en las instrucciones por parte de los

equipos de terreno, entre otras. Todas estas situaciones podrían tener un cambio en el modo en el que se registra la información, haciendo que los nuevos datos se distancien de los datos de entrenamiento

Una segunda alternativa sería implementar un sistema automatizado que genere una métrica que compare las características de los datos de entrenamiento con aquellos para los que se requiere una predicción. Esta métrica opera como una *proxy* de precisión, lo cual permite moniteorear el rendimiento del modelo por medio de una variable que no es la preción, pero que sí se relaciona con ella.

Independientemente de la estrategia para hacer el seguimiento del modelo, la acción esperada es una revisión del proceso de entrenamiento. En ciertas ocasiones el problema podría solucionarse con ciertos ajustes en el procedimiento, pero si el problema se explica por un cambio importante en los datos nuevos respecto a los de entrenamiento, es decir, por una pérdida de representatividad de estos últimos, la alternativa más adecuada es la adición de un conjunto nuevo de datos etiquetados manualmente.

Generación de los conjuntos de datos de entrenamiento

Principales fuentes de información y codificación

Dado que el objetivo del proyecto era generar conjuntos de datos provenientes de encuestas realizadas mediante CAPI, las principales fuentes de información corresponden a dicha modalidad. Dos son las fuentes de información más importantes: 1) coyuntura ENE entre los meses de junio y noviembre de 2020 y 2) encuesta Piloto EPF 2021. Una tercera fuente de información muy marginal [PONER DATO] corresponde a la coyuntura ENE del año 2018, pero solo para algunas categorías de CAENES, escasamente representadas en la base de datos construida inicialmente.

Respecto al origen de la codificación, la mayor parte de esta fue realizada por un equipo de codificadores que apoyaron las actividades del Proyecto Estratégico de Servicios Compartidos. Dicho equipo etiquetó datos provenientes de la coyuntura ENE entre los meses de junio y noviembre de 2020. Una segunda fuente de etiquetado corresponde al control de calidad realizado por el Departamento de Estudios del Trabajo (DET) en el marco de la coyuntura de la Encuesta de Empleo. Debido a que estos datos ya habían sido revisados por un analista entrenado, pasaron directamente a formar parte del dataset de entrenamiento. En tercer lugar, en una proporción mucho menor, fueron agregados algunos datos de la coyuntura ENE del año 2018, ya mencionados previamente.

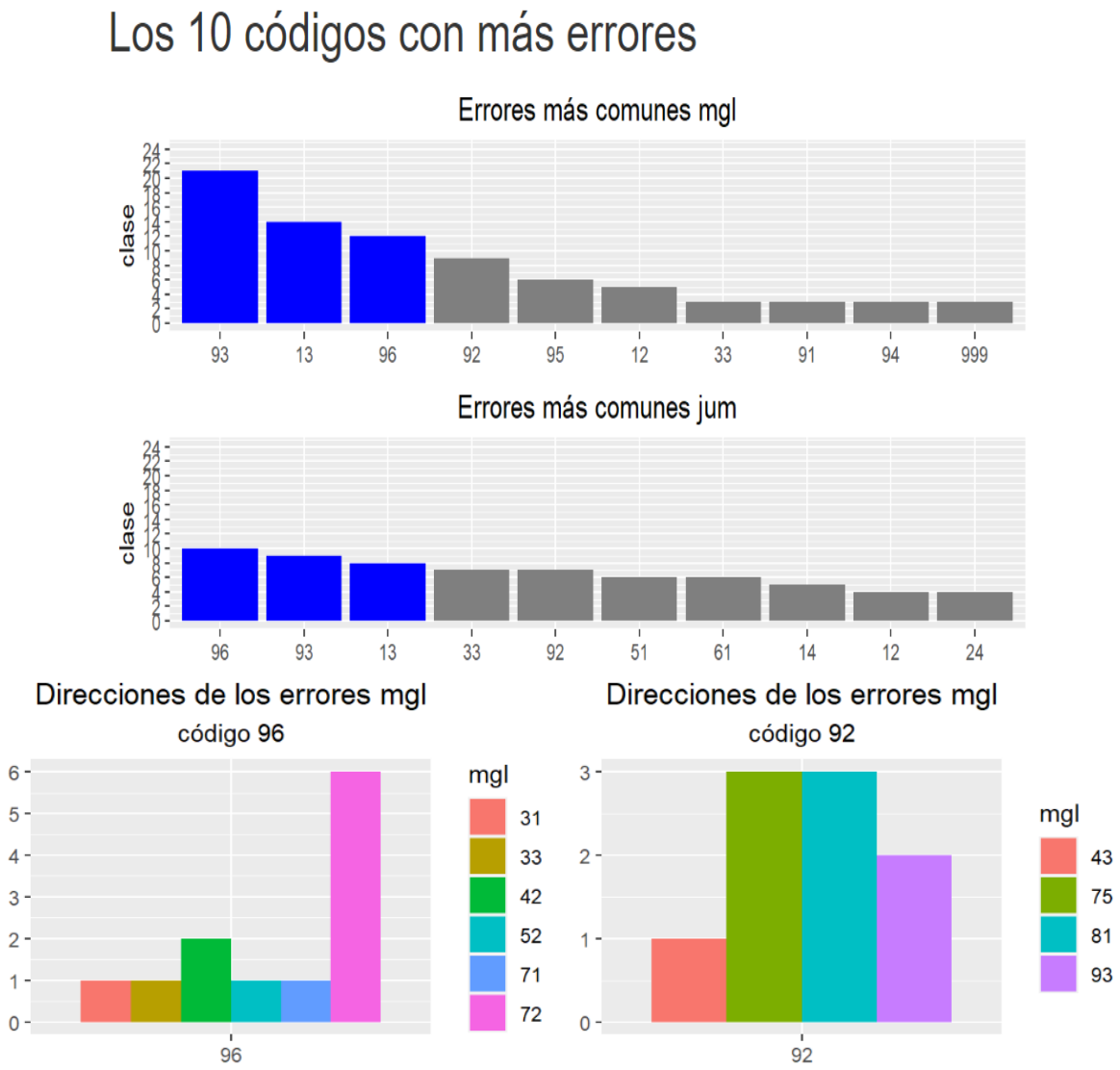
Proceso de codificación

Codificación cruzada

Con el objeto de alcanzar una codificación de alta calidad se implementó un sistema de doble etiquetado. Así, al principio del proceso cerca de un 90% de los datos fue codificado por 2 analistas diferentes, de modo de conseguir una mayor seguridad respecto al código asignado para cada glosa. Cuando se producía una diferencia en el código asignado, un tercer analista más calificado, discernía cuál era el código final. Es importante mencionar que el porcentaje de doble codificación fue disminuyendo cuando se detectó que la cantidad de registros con discrepancias comenzó a decaer. La decisión de diminuir el porcentaje de doble codificación obedece a motivos de costos, ya que para cumplir con los objetivos del proyecto, se hizo necesario aumentar la velocidad de codificación.

Vale la pena mencionar que durante todo el proceso de codificación se contó con el apoyo del equipo de Nomenclaturas, el cual revisaba una muestra mensual de aproximadamente 150 registros. A partir de dicha revisión, se retroalimentaba al equipo de codificadores, tanto en CAENES como en CIUO. Esto permitió rectificar ciertos criterios y revisar algunos registros que habían sido codificados sobre la base de reglas mal implementadas. Adicionalmente, la auditoría se utilizó para mejorar la calidad de la codificación, ya que para todos los casos que fueron seleccionados para auditar, el código asignado por el equipo del proyecto estratégico fue reemplazado por el de Nomenclaturas.

Las siguiente figuras muestra fragmentos del tablero utilizado para retroalimentar a los codificadores sobre la base de la información proporcionada por la auditoría del equipo de Nomenclaturas. Mediante esta herramienta se informaba cuáles eran los códigos que tenían mayor número de errores y cuál era la dirección de dichos errores. Esto último fue de gran ayuda para incorporar los criterios del clasificador, ya que muchas veces existen sutilezas que determinan la pertenencia de una glosa a uno u otro subgrupo principal.



Revisión de códigos complejos

Sobre la base de la información recolectada de la codificación cruzada y de las auditorías, se tomó la decisión de llevar a cabo una revisión masiva de algunos códigos más complejos. Para el caso de CIUO, la unidad de Nomenclaturas revisó 2.000 registros.

En el caso de CAENES se realizaron dos revisiones. En primer lugar, 15.318 registros pertenecientes a códigos complejos. En segundo lugar, se revisaron 5.904 registros que cumplían con la característica de tener glosas idénticas, pero con códigos diferentes.

Auditorías finales

Una vez finalizada la revisión de glosas complejas, se llevó a cabo una auditoría que contempló 3.000 registros: 1.500 para CAENES y 1.500 para CIUO. El resultado de esta auditoría final muestra que para los registros de CIUO codificados por el equipo de Servicios Compartidos (SSCC) existe una coincidencia de 91,1% respecto a lo señalado por Nomenclaturas. En el caso de los registros provenientes del control de calidad de la ENE el porcentaje de acierto es notoriamente menor y, justamente, por ello dichos registros fueron excluidos del proceso de entrenamiento. Se volverá sobre este punto en uno de los siguientes apartados.

[AGREGAR NÚMERO]

Descripción de los sets de entrenamiento

CAENES

En el caso de CAENES, el etiquetado de los datos proviene de 3 fuentes. La primera de dichas fuentes corresponde al trabajo realizado por un equipo de codificadores en el marco de las actividades del Proyecto Estratégico de Servicios Compartidos. Dicho equipo etiquetó datos provenientes de la coyuntura ENE entre los meses de junio y noviembre de 2020.

La segunda fuente de etiquetado corresponde al control de calidad realizado por el Departamento de Estudios del Trabajo (DET) en el marco de la coyuntura de la Encuesta de Empleo. Debido a que estos datos ya habían sido revisados por codificadores entrenados, pasaron directamente a formar parte del dataset de entrenamiento.

En tercer lugar, en una proporción mucho menor, fueron agregados algunos datos de la coyuntura ENE del año 2018, con el objetivo de aumentar algunas categorías con poca prevalencia.

Cabe mencionar que de las tres fuentes de información, la más importante proviene del etiquetado realizado por el Proyecto Estratégico de Sevicios Compartidos.

A continuación se describen las variables más relevantes del archivo:

- idrph: identificador de persona
- glosa_caenes: descripción de la actividad económica
- cod_final: código caenes a dos dígitos
- origen: procedencia del etiquetado. Las categorías son ene y sscc (Servicios Compartidos)
- levantamiento: indica si el levantamiento fue mediante papel o dispositivo electrónico. Las categorías son: papi y capi
- tiene_auditoria: fue auditado por el equipo de Nomenclatura. Las categorías son 0 y 1.
- tiene_rev_cruzada: el caso tuvo revisión cruzada. Las categorías son 0 y 1.
- variable: indica la pregunta del cuestionario de la cuál fue obtenido el dato. Las categorías son c5, c9 y d5.
- Debido a que cada persona puede tener más de un registro, la manera de generar un identificador único es mediante las columnas idrph, mes y variable.

CIUO

En el caso de CIUO, el etiquetado de los datos proviene de 2 fuentes. La primera de ellas corresponde al trabajo realizado por un equipo de codificadores en el marco de las actividades del Proyecto Estratégico de Servicios Compartidos. Dicho equipo etiquetó datos provenientes de la coyuntura ENE entre los meses de junio y noviembre de 2020, y en menor medida, datos de la encuesta piloto EPF, realizada durante el segundo semestre de 2020.

La segunda fuente de etiquetado corresponde al control de calidad realizado por el Departamento de Estudios del Trabajo (DET) en el marco de la coyuntura de la Encuesta de Empleo. Debido a que estos datos ya habían sido revisados por codificadores entrenados, pasaron directamente a formar parte del dataset de entrenamiento.

A continuación, se describen las variables más relevantes del archivo:

- idrph: identificador de persona
- b1_1: oficio
- b1_2: tarea
- cod_final: código asignado a 2 dígitos
- encuesta: ENE o piloto EPF
- origen: procedencia del etiquetado. Las categorías son ene y sscc (Servicios Compartidos)
- levantamiento: indica si el levantamiento fue mediante papel o dispositivo electrónico. Las categorías son: papi y capi
- tiene_auditoria: fue auditado por el equipo de Nomenclatura. Las categorías son 0 y 1.
- tiene_rev_cruzada: el caso tuvo revisión cruzada. Las categorías son 0 y 1.
- Debido a que cada persona puede tener más de un registro, la manera de generar un identificador único es mediante las columnas idrph, mes.

Caracterización de los datos de entrenamiento

Dataset CAENES

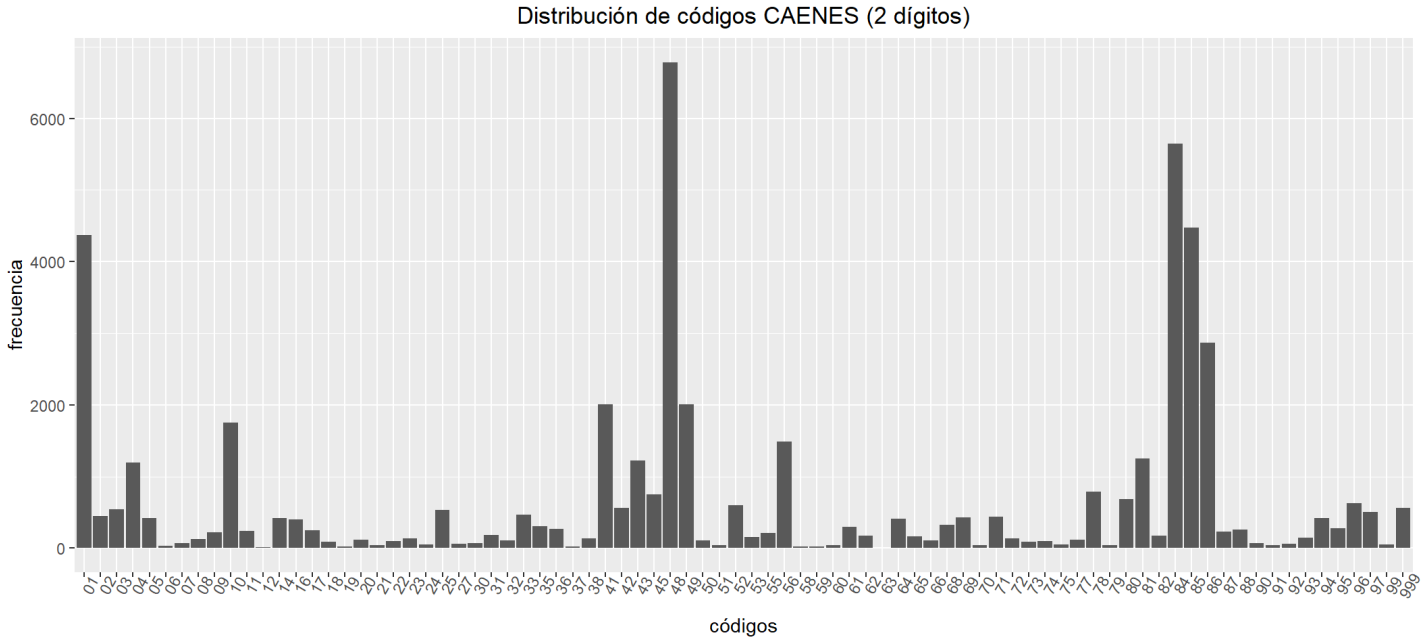
Para el caso de CAENES, el set de datos está conformado por 51.352 registros, pero es importante considerar que 139 glosas solo contienen el valor “88”. Sin considerar dichas filas, el total de glosas válidas es 51.213. En lo sucesivo se presentarán datos excluyendo dichas glosas.

Respecto al origen de la codificación, un 23,3% proviene del control de calidad de la coyuntura ENE y un 76,7% corresponde a glosas etiquetadas en el marco del Proyecto Estratégico de Servicios Compartidos. Vale la pena mencionar que en el caso de los datos provenientes del control de calidad ENE, no hubo una doble codificación, ya que se asumió que por las características del procesamiento de la ENE, no era necesario llevar a cabo un nuevo chequeo, por lo que aquellos datos pasaron inmediatamente a formar parte del set de entrenamiento final.

Origen de la codificación

origen	frecuencia	porcentaje
ene	11.921	23,3
sscc	39.292	76,7

El siguiente gráfico muestra la distribución de la variable CAENES a dos dígitos. Vale la pena mencionar que los registros con código 999 corresponden a casos en los que la información contenida en las glosas era insuficiente para etiquetar a dos dígitos. Este código tiene una incidencia de 1,1% respecto al total.



La siguiente tabla muestra algunos ejemplos de glosas asociadas al código 999. Se puede observar que en algunos casos se podría asignar un código a un dígito, sin embargo, el nivel de detalle exigido por el clasificador no permite codificar a dos dígitos.

Códigos 999 (algunos ejemplos)

glosa_caenes
contratista en la construccion
ong. organización no gubernamental
telecomunicaciones
servicio de alimentos
servicios integrales para la mineria
prestadora de servicios a mineras
servicios generales
contruccion

Finalmente, se muestra la cantidad y porcentaje de registros provenientes de levantamiento CAPI y PAPI. Tal como se señala en uno de los apartados iniciales, una preocupación a lo largo de este trabajo ha sido generar conjuntos de entrenamiento que consideren datos generados a partir de dispositivos móviles. En ese sentido, cerca de un 100% corresponde a levantamientos via tablet, pero dado que algunas categorías presentaban una escasa presencia, se tomo la decisión de incorporar algunas glosas del levantamiento de 2018, de modo de abultar dichas categorías y mejorar el proceso de entrenamiento.

Procedencia CAPI y PAPI

levantamiento	frecuencia	porcentaje
capi	50.749	99.1
papi	464	0.9

Dataset CIUO

Para el caso de CIUO, el set de datos está conformado por 33.846 registros, pero al igual que en el caso de CAENES, existen glosas que presentan valor 88. La siguiente tabla muestra la cantidad de registros con valores 88 en las variables oficio y tarea. Además, se indica el número de registros en los que tanto oficio como tarea tiene valor 88. Se observa que el total de registros válidos es 33.823.

Cantidad valores perdidos

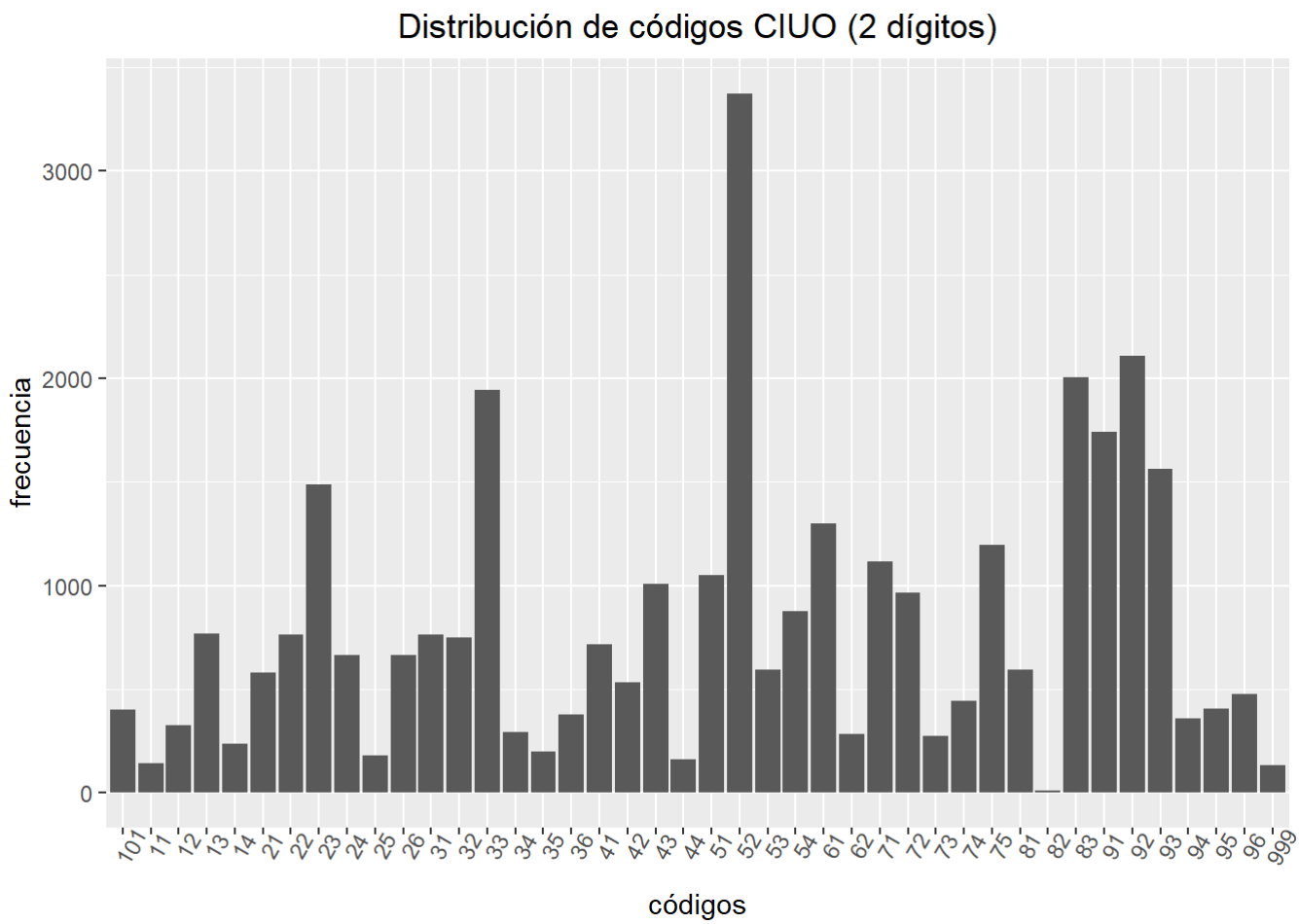
missing	frecuencia	porcentaje
missing oficio	1	0,0
missing oficio y tareas	4	0,0
missing tareas	18	0,1
no missing	33.823	99,9

Respecto al origen de la codificación, la siguiente tabla da cuenta de que el 63,8% de los registros fueron codificados en el marco del Proyecto Estratégico de Servicios Compartidos.

Origen de la caodificación

origen	frecuencia	porcentaje
ene	12.250	36,2
sscc	21.573	63,8

El siguiente gráfico muestra a distribución de la variable CIUO a dos dígitos. Al igual que en el caso de CAENES, los valores 999 corresponden a glosas, cuya información no permitía llevar a cabo una codificación adecuada a dos dígitos. La categoría 999 corresponde al 0,4 del total.



La siguiente tabla muestra algunos ejemplos de glosas que no pudieron ser codificadas a dos dígitos debido a la falta de información.

Códigos 999 CIUO (algunos ejemplos)

b1_1	b1_2
informatico	se encarga de
auxiliar de escenario	preparar y coordinar los escenarios para actividades, seminarios, reuniones
venta de alimentos	preparar, vender alimentos y atención público
tecnica en sanitizacion	tramites de empresa
empresario	compra y venta de todo tipo de mercadería desde camiones.
supervisor en telecomunicaciones	a cargo de area telefonia
supervisora	supervisar en terreno el desempeño de trabajadores y funcionamiento de maquinarias
enfermera naval	realuza atención , control y derivación de pacientes en servicio de urgencias

Descripción de los modelos

Durante el proceso de entrenamiento se llevó a cabo una serie de pruebas, con el objeto de encontrar el mejor modelo posible. A grandes rasgos, se utilizaron variantes para 2 partes del entrenamiento: 1) metodología para convertir textos en vectores y 2) arquitectura del algoritmo para hacer el entrenamiento. Si bien existen otras partes del entrenamiento que pueden impactar en el resultado final, las 2 seleccionadas son las que probablemente tengan un mayor efecto. A raíz de ello, se priorizó por ellas.

Metodología para vectorizar textos

Para que los datos de texto sean procesados por cualquier algoritmo, se requiere que estos sean convertidos a algún tipo de representación numérica. Para llevar a cabo esta trasformación existen diferentes estrategias. Dos de las más importantes son *bag of words (BOW)* y *word embeddings*. En el caso de BOW es posible encontrar una serie de variantes, desde el enfoque más simple consistente en identificar la presencia o ausencia de una palabra, hasta representaciones más complejas, como tf-idf, que buscan ponderar las palabras de acuerdo a la importancia que estas tienen en un determinado corpus. La segunda vertiente, *word embeddings*, consiste en utilizar representaciones vectoriales de cada palabra, provenientes de un entrenamiento realizado con grandes volúmenes de texto. Típicamente, se utilizan vectores de 300 dimensiones entrenados con toda la información existente en wikipedia y/o otras fuentes similares. Dado que cada palabra de un texto es representada por un vector, se hace necesario alguna medida sintética que permita llegar a una representación para el texto completo. Para ello, existen distintas estrategias y lo que se utilizó en este trabajo fue la media de cada una de las 300 dimensiones. Así, cada glosa es representada por un vector de 300 dimensiones. Dicho vector contiene la información “resumida” de las palabras que componen cada glosa.

En el marco de este trabajo se llevaron a cabo pruebas con ambos enfoques.

- Estrategia BOW: Dentro de la estrategia BOW se utilizaron 2 variantes. En primer lugar, se construyeron vectores utilizando tf-idf. Ello permite extraer de mejor manera el “significado” de cada palabra, que utilizando únicamente la presencia o ausencia. De este modo, un texto queda representado por un vector que tiene tantas dimensiones como palabras diferentes existan en el corpus. Dado que un texto por lo general contiene solo una pequeña fracción del total de palabras del vocabulario, el resultado corresponde a una matriz en la que la mayoría de sus elementos es 0. Típicamente, seguir esta metodología implica construir vectores con varios miles de dimensiones, lo cual supone un alto costo de procesamiento, pero

dado que la mayor parte de los elementos corresponde a valores 0, es posible utilizar algunas estrategias para disminuir de manera importante el tiempo de ejecución.

La segunda de BOW variante utilizada en este trabajo consiste en asociar cada palabra diferente del corpus a un índice y luego representar cada texto como una secuencia de índices. Esto quiere decir que un texto con 5 palabras quedará representado por un vector de 5 dimensiones. Es importante mencionar que dado que los textos no tienen la misma cantidad de palabras, se debe recurrir a alguna estrategia para que todos los vectores terminen con la misma cantidad de elementos. Para ello, se utiliza un procedimiento denominado *padding*, que consiste en fijar un valor para el largo de los vectores y completar con ceros todos los vectores que tengan un número menor a dicho valor máximo.

Cabe señalar que en ambos enfoques de BOW la "semántica" de las palabras proviene de la información presente en el set de datos que se utiliza para el entrenamiento. En ese sentido, no se recurre a información externa, lo cual si ocurre en el enfoque basado en *word embeddings*.

- Estrategia *word embedding*: Como ya se ha mencionado, *word embeddings* es una manera de construir representaciones de palabras por medio de un proceso de aprendizaje en el cual se utilizan grandes volúmenes de texto. Así, una red neuronal es entrenada para que, a partir de una palabra, sea predicho el contexto de la misma o, inversamente, el entrenamiento puede realizarse para que a partir de un contexto, se prediga una palabra. Cuando este proceso se hace con una cantidad de datos suficientes, es posible que la red "aprenda" representaciones de las palabras que capturan su semántica.

En el marco de este trabajo se utilizó un modelo entrenado por Jorge Perez (https://github.com/dccuchile/spanish-word-embeddings/blob/master/emb-from-suc.md). El modelo contiene vectores de 300 dimensiones y su entrenamiento se hizo mediante el algoritmo FastText.

Arquitecturas probadas

Durante la etapa de pruebas, se llevaron a cabo ejercicios con 2 arquitecturas de redes neuronales. Dichas arquitecturas se probaron con las dos estrategias de vectorización mencioandas más arriba, a partir de lo cual se originan 4 modelos diferentes.

Las arquitecturas consideradas fueron *feed-forward* y *gated recurrent unit* (GRU) . La arquitectura de red feed-forward es la configuración más sencilla de redes neuronales y se incorpora en este trabajo con el objeto de contar con una medida base contra la cual contrastar otras arquitecturas de redes más complejas. Por su parte, las redes GRU son una variante de la arquitectura LSTM (*Long short-term memory*), pero con una estructura más simple y, por ende, con una menor cantidad de parámetros a estimar.

Ambas arquitecturas (LSTM y GRU) comparten la característica de permitir que parte de la información inicial de una secuencia de datos sea traspasada hacia los estados de más adelante, combatiendo el problema de la "falta de memoria" de largo plazo. En ese sentido, la arquitectura GRU es sumamente útil para procesar datos de texto, ya que el lenguaje humano puede ser entendido como una secuencia de elementos concatenados, en los que cierto orden genera sentido.

Resultados del entrenamiento

Los cuadros siguientes resumen los resultados obtenidos en el set de testeo para cada una de las estrategias y desagregaciones realizadas. Es importante mencionar que los modelos fueron entrenados con un subconjunto de los datos de entrenamiento, ya que solo se utilizaron aquellos **etiquetados por codificadores de SSCC** [3].

En el caso de CAENES no se advierten diferencias importantes entre las distintas extrategias elegidas, sin embargo, vale la pena mencionar que tanto a uno como a dos dígitos, la vectorización mediante secuencias (seq_1d y seq_2d) muestra un desempeño ligeramente más alto que los demás modelos. Esto se debe a que en la extracción de características para esa estrategia, se utilizó la información de la variable CISE. Pese a ello, se recomienda que el modelo basado en *word embeddings* sea utilizado de manera productiva, ya que el hecho de enriquecer la descripción de los textos mediante un modelo entrenado con grandes cantidades de texto, hace suponer que las predicciones sean menos sensibles a cambios en la manera de registrar la información [4].

En el caso de CIUO es un poco más evidente que el modelo que utiliza *word embeddings* presenta un mejor rendimiento que los demás, por lo que, nuevamente, se sugiere que sea utilizado de manera productiva.

Resultados CAENES 1 dígito

modelo	acc	macro	micro	weighted
seq_1d	0.9384	0.8757	0.9384	0.9386
tfidf_1d	0.9327	0.8658	0.9327	0.9330
emb_simple_1d	0.9274	0.8641	0.9274	0.9280
emb_gru_1d	0.9327	0.8694	0.9327	0.9328

Resultados CAENES 2 dígitos

modelo	acc	macro	micro	weighted
seq_2d	0.9075	0.7536	0.9075	0.9083
tfidf_2d	0.9076	0.7579	0.9076	0.9081
emb_simple_2d	0.9021	0.7631	0.9021	0.9039
emb_gru_2d	0.9048	0.7630	0.9048	0.9052

Resultados CIUO 1 dígito

modelo	acc	macro	micro	weighted
seq_1d	0.8858	0.8599	0.8858	0.8855
tfidf_1d	0.8684	0.8362	0.8684	0.8686
emb_simple_1d	0.8793	0.8519	0.8793	0.8807
emb_gru_1d	0.8989	0.8796	0.8989	0.8990

Resultados CIUO 2 dígitos

modelo	acc	macro	micro	weighted
seq_2d	0.8456	0.7249	0.8456	0.8476
tfidf_2d	0.8412	0.7355	0.8412	0.8431
emb_simple_2d	0.8324	0.7220	0.8324	0.8348
emb_gru_2d	0.8526	0.7364	0.8526	0.8543

Modelos en producción

El presente proyecto pone a disposición de los usuarios el código fuente (scripts en R y Python) y los datos necesarios para reproducir los resultados presentados más arriba, de manera tal de transparentar las decisiones metodológicas y abrir el espacio para futuras mejoras. Ahora bien, para reproducir completamente los resultados es necesario que el usario cuente con una serie de componentes, lo cual no siempre es una tarea sencilla. De manera esquemática, se presentan las principales herramientas para ejecutar las rutinas:

- R y RStudio
- Python
- Reticulate
- Keras/Tensorflow

Además de lo anterior, es necesario considerar que el entrenamiento de los modelos fue realizado en un sistema operativo Linux y hasta el momento se ha probado que las rutinas se ejecuten correctamente en CENTOS y Ubuntu, pero no en Windows. Adicionalmente, se utilizó un *virtual env* de Python, para facilitar la vinculación entre R y Python. Las dependencias señaladas más arriba, sumado a ciertas decisiones (sistema operativo Linux y vinculación de R con Python) suponen una barrera de entrada importante para que los modelos sean utilizados con fluidez.

Con el objeto de que la codificación automática sea utilizada por la mayor cantidad de usuarios posible, se pone a diposición una API (*Application Programming Interface*), que posibilita que los modelos de clasificación sean utilizados de manera relativamente sencilla. Una API permite que un usuario se comunique con un servicio, sin la necesidad de conocer los detalles de la implementación que hay detrás de dicho servicio. Basta con que una solicitud (*request*) sea realizada con cierta estructura, para que la API retorne un resultado, que también cumple con cierta estructura. En ese sentido, la API que se pone a disposición permite que los modelos sean utilizados, sin que el usuario cuente con ninguno de los requerimientos mencionados más arriba, disminuyendo de manera significativa las barreras de entrada.

En ese sentido, desde el punto de vista de un analista, la única herramienta que se requiere para poder utilizar los modelos es alguna librería que permita hacer solicitudes a un servidor. Independientemente de la herramienta utilizada para hacer la solicitud, la API retornará la predicción del modelo en formato json, el cual es fácilmente manipulable en casi cualquier lenguaje de programación.

A continuación se presenta un ejemplo de cómo interactuar con la API mediante el paquete `httr` de `R`. El primer parámetro del request corresponde al servidor y al *endpoint* para la predicción. Dado que no es deseable migrar la API de un servidor a otro, este parámetro debería quedar relativamente estable. En segundo lugar, debe indicarse que se trata de un archivo json, lo cual también debiese ser algo que no sufra variaciones. Dentro del cuerpo del *request* debe indicarse cuáles son los textos para los cuales se requiere una predicción. En este caso se están enviando al servidor las 10 primeras glosas de la columnaa *glosa_caenes*. Adicionalmente, debe indicarse si los datos corresponden a ocupación o actividad económica. Finalmente, se debe indicar si la predicción es a uno o dos dígitos.

```
# Predecir a un dígito
library(httr)
library(feather)
caenes <- read_feather("src/data/split_train_test/test.feather")
request <-  httr::POST("http://143.198.79.143:8080/predict",
                 encode = "json",
                 body =  list(text = caenes$glosa_caenes[1:10],
                              classification = "caenes",
                              digits = 1)
)

httr::status_code(request)
response <- httr::content(request)
```

Conclusión

El presente proyecto ha tenido el objetivo de generar datos de entrenamiento y modelos para ser utilizados libremente por la institución. En ese sentido, este esfuerzo se enmarca en un camino que la institución viene recorriendo desde hace algunos años hacia la automatización de procesos de codificación. En este camino, el enfoque de *machine learning* ha generado resultados positivos y es de esperar que su utilización siga expandiéndose con el paso del tiempo. Ahora bien, al mismo tiempo en que se llevan a cabo avances en el campo de la inteligencia artificial, surgen nuevos desafíos institucionales, que no deben ser soslayados.

En primer lugar, es sumamente importante considerar que la continuidad operativa de los modelos puestos a disposición requiere de un equipo que pueda hacerse cargo del monitoreo y actualziaciones que el sistema requiera. En ese sentido, el equipo de Nomenclaturas surge como el más idóneo para llevar a cabo dicha tarea, ya que en él reside la responsabilidad de implementar los estándares internacionales de los clasificadores a la realidad nacional.

A lo largo de este documento se ha mencionado la necesidad de monitoreo constante de los modelos. Tal como se ha señalado antes, una estrategia basada en una revisión manual por parte de analistas especializados es efectiva, sin embargo, conlleva costos elevados. En ese sentido, una estrategia automática de seguimiento constituiría una herramienta útil para mejorar y facilitar la continuidad operativa de los modelos en producción. Esto podría ser pensado como un mecanismo automático, que genere alertas cuando exista probabilidad de que un modelo comience a fallar.

---

1. En el documento metodológico “Sistema de clasificación y codificación automática en la Encuesta Nacional de Empleo” se describe la metodología que actualmente utiliza la ENE para codificar CAENES y CIUO. Disponible en https://www.ine.cl/docs/default-source/ocupacion-y-desocupacion/metodologia/espanol/documento-sistema-de-clasificaci%C3%B3n-y-codificaci%C3%B3n-autom%C3%A1tica-(mayo-2019).pdf?sfvrsn=ceea6423_3 (https://www.ine.cl/docs/default-source/ocupacion-y-desocupacion/metodologia/espanol/documento-sistema-de-clasificaci%C3%B3n-y-codificaci%C3%B3n-autom%C3%A1tica-(mayo-2019).pdf?sfvrsn=ceea6423_3)↩
2. Si comienza a generarse una distancia considerable entre los datos de entrenamiento y aquellos para los que se pretende hacer una predicción, es posible que el rendimiento estimado inicialmente vaya disminuyendo progresivamente. Es deseable, entonces, contar con herramientas que permitan identificar cuándo la predicción del modelo comienza a disminuir. Ahora bien, la identificación de dicho punto no es trivial. La manera más sencilla, pero al mismo tiempo más costosa, es la revisión periódica de las predicciones del modelo por parte de un analista altamente entrenado en el sistema de clasificación. Con ello se busca comparar la predicción del modelo con algo que en el contexto de *machine learning* se suele llamar *ground truth*, es decir, con una codificación considerada correcta. En este caso, se asume que los códigos asignados por un analista experto son nuestra *ground truth*. Si una estrategia como esta se realiza con la periodicidad adecuada, es posible identificar el momento en el cual la precisión del modelo comienza a disminuir. La limitación más importante es que se requiere contar con recursos humanos altamente especializados para esta tarea.↩
3. La variable origen permite filtrar los datos etiquetados por el equipo de SSCC.↩
4. Para mayores detalles sobre la construcción de *word embeddings*, ver el trabajo de Jorge Perez Rojas [https://github.com/dccuchile/spanish-word-embeddings (https://github.com/dccuchile/spanish-word-embeddings)]↩