

Predicción del tiempo entre rehospitalizaciones

**Ana Isabel del Val Franco**

Máster en Bioestadística y Bioinformática

Area 3, Aula 1

**Consultor: Albert Pla Planas**

**Profesor responsable: Ferran Prados Carrasco**

05/01/2021

  
Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](http://creativecommons.org/licenses/by-nc-nd/3.0/es/)

**Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)**

**A) Creative Commons:**

  
Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](http://creativecommons.org/licenses/by-nc-nd/3.0/es/)

  
Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](http://creativecommons.org/licenses/by-nc-sa/3.0/es/)

  
Esta obra está sujeta a una licencia de Reconocimiento-NoComercial [3.0 España de Creative Commons](http://creativecommons.org/licenses/by-nc/3.0/es/)

  
Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada [3.0 España de Creative Commons](http://creativecommons.org/licenses/by-nd/3.0/es/)

  
Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual [3.0 España de Creative Commons](http://creativecommons.org/licenses/by-sa/3.0/es/)

  
Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](http://creativecommons.org/licenses/by/3.0/es/)

**B) GNU Free Documentation License (GNU FDL)**

Copyright © AÑO TU-NOMBRE.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

**C) Copyright**

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

**FICHA DEL TRABAJO FINAL**

|  |  |
| --- | --- |
| **Título del trabajo:** | *Predicción del tiempo entre rehospitalizaciones* |
| **Nombre del autor:** | *Ana Isabel del Val Franco* |
| **Nombre del consultor/a:** | *Albert Pla Planas* |
| **Nombre del PRA:** | *Ferran Prados Carrasco* |
| **Fecha de entrega (mm/aaaa):** | 01/2021 |
| **Titulación::** | *Máster en Bioinformática y Bioestadística* |
| **Área del Trabajo Final:** | *Area3, Aula 1* |
| **Idioma del trabajo:** | *Castellano* |
| **Palabras clave** | *EHR, W2V, Rehospitalización* |
| **Resumen del Trabajo (máximo 250 palabras):** *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.* | |
| **Objetivo:**  Predicción del tiempo entre rehospitalizaciones de pacientes registrados en la base de datos MIMIC-III, basado en la severidad de la enfermedad, diagnósticos, prescripciones médicas y notas del doctor.  **Contexto:**  El dataset EHR está organizado por hospitalizaciones, no por pacientes, asumiendo independencia entre elementos. La distribución de los tiempos de admisión y de alta están descritos por 58976 hospitalizaciones de 46520 pacientes. No todas ellas contienen notas del doctor, resultando en 39166 hospitalizaciones de 30423 pacientes para el diseño del modelo, y 39641 filas.  **Metodología:**  La severidad depende del número de admisiones, número de estancias en la UCI y el desenlace de la enfermedad (vida o fallecimiento). Hay dos valores posibles: Bajo y Alto.  Los diagnósticos están originalmente expresados en código ICD9. Se considera que la misma hospitalización puede ser diagnosticada con varias enfermedades. Únicamente los diagnósticos más frecuentes se consideran en el modelo, incluyendo la Artritis Reumatoide.  Las prescripciones administradas en cada admisión están también filtradas por los valores más frecuentes con la finalidad de evitar complejidad.  Tanto los diagnósticos como las prescripciones han sido codificadas y transformadas a columnas siguiendo la técnica one-hot encoding.  Las notas del doctor contienen información sobre la historia clínica de los pacientes y la información en el momento del alta. Estas notas han sido modeladas implementando la técnica Word2Vec, transformando el texto en vectores de 96 elementos. Estos vectores, a su vez, han sido transformados también en columnas del conjunto de datos.  Los modelos son entrenados con la técnica cross-validation. Los datos etiquetados conforman el 23.71% de los datos disponibles, usando el 80% para su entrenamiento.  Los modelos implementados son de clasificación: Random Forest, NaiveBayes, Support Vector Machine y árboles de decisión.  **Resultados:**  https://drive.google.com/drive/folders/1m8M-MsGO0TiDU-f9LcVQErQTc5X36DVE  **Conclusiones:**  El acceso a un dataset real es largo y tedioso. | |
| **Abstract (in English, 250 words or less):** | |
| **Objective:**  Prediction of time between rehospitalizations of patients registered in MIMIC-III database, based on the severity of the illness, the diagnoses, the prescriptions and the doctor notes.  **Context:**  The EHR dataset is organized by hospitalization, not by patient, assuming independence between items. The distribution of the admission and discharge times are described in 58976 hospitalizations of 46520 patients. Not all of them contain doctor notes, resulting in 39166 hospitalizations of 30423 patients for the model design, and 39641 rows.  **Methodology:**  The severity depends on the number of admissions, number of ICU stays and the outcome of the illness (alive or death). There are two possible values: Low and High.  The diagnoses are originally expressed in ICD9 code. It is considered that the same hospitalization can be diagnosed with several illnesses. Only the most frequent diagnoses are considered in the model, including Rheumatoid Arthritis.  The prescriptions administered in each admission are also filtered by the most frequent values in order to avoid complexity.  Both the diagnoses and prescriptions have been encoded and transformed to columns following the one-hot encoding technique.  The doctor notes contain information about the clinical history of the patients and the information at the time of discharge. These notes have been modeled implementing Word2Vec technique, transforming the text to vectors of 96 elements. These vectors, in turn, have also been transformed to columns of the dataset.  The models are trained based on cross-validation technique. The labeled data conform the 23.71% of the available data, using 80% for training.  The models implemented are classifiers: Random Forest, NaiveBayes, Support Vector Machine and decision trees.  **Results:**  https://drive.google.com/drive/folders/1m8M-MsGO0TiDU-f9LcVQErQTc5X36DVE  **Conclusions:**  The access to a real dataset could be long and tiresome. | |

**Índice**

[**1. Introducción**](#_heading=h.48js7y6804t2) **7**

[1.1 Contexto y justificación del Trabajo](#_heading=h.j9h0b7b7d86x) 7

[1.2 Objetivos del Trabajo](#_heading=h.ct1vmfv9dahf) 8

[1.3 Enfoque y método seguido](#_heading=h.oe5gy3mhsfmi) 8

[1.4 Planificación del Trabajo](#_heading=h.ea96c088bg9l) 9

[1.5 Breve sumario de productos obtenidos](#_heading=h.tr21nowjhu0a) 10

[1.6 Breve descripción de los otros capítulos de la memoria](#_heading=h.81ntcasg1952) 11

[**2. Desarrollo**](#_heading=h.3yxaeau1vz3m) **12**

[2.1 Arquitectura de tablas](#_heading=h.l6pmc6bb3n3s) 12

[2.2 Conexión con la instancia ec2 de AWS](#_heading=h.mg3dft6vd89x) 16

[2.3 Estudio longitudinal de la población](#_heading=h.mg3dft6vd89x) 16

[2.4 Severidad](#_heading=h.mg3dft6vd89x) 17

[2.5 Diagnósticos](#_heading=h.mg3dft6vd89x) 17

[2.6 Prescripciones](#_heading=h.mg3dft6vd89x) 18

[2.7 Notas del doctor](#_heading=h.mg3dft6vd89x) 19

[2.8 Tiempo entre rehospitalizaciones](#_heading=h.8fdxtets7gn) 19

[2.9 Organización del dataset](#_heading=h.mg3dft6vd89x) 20

[2.10 Modelos](#_heading=h.mg3dft6vd89x) 20

[**3. Conclusiones**](#_heading=h.5rvdbqgg8bkf) **22**

[**4. Glosario**](#_heading=h.ooytbcoysa8s) **23**

[**5. Bibliografía**](#_heading=h.rdw4ptgwyh7l) **24**

[**6. Anexos**](#_heading=h.4383unp3xed4) **25**

[6.1 Amazon EC2 key pairs and Linux instances [11]](#_heading=h.d3vyug1330q0) 25

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

* 1. Los historiales médicos electrónicos (EHR) son una fuente muy valiosa de información para ayudar en la investigación y la toma de decisiones clínicas. Concretamente, las conclusiones de los historiales de datos reales, aunque más difícilmente accesibles, pueden extrapolarse al mundo científico, no siendo así para el caso de los datos sintéticos, que aun siendo su acceso más sencillo, han sido creados sin mucha transparencia para fines concretos y difícilmente adecuados para otros fines.

1. MIMIC-III (Medical Information Mart for Intensive Care III) es una base de datos médicos muy extensa que contiene datos de más de 40.000 pacientes, los cuales estuvieron ingresados en unidades de cuidado del “Beth Israel Deaconess Medical Center” entre 2001 y 2012. Esta base de datos contiene información demográfica, medidas de signos vitales, resultados de pruebas de laboratorio, procedimientos, medicaciones, notas del doctor, información de los cuidadores, reportes de imagen, … Es accesible bajo petición justificada y previa cumplimentación de un curso acerca de “Human research subject protections and HIPAA regulations”.
   1. En la investigación de recaídas afectan diversos factores, los cuales, no siempre están almacenados en una única fuente que permita cruzar datos y extraer conclusiones. Este proyecto ayuda a poner estos factores en común pudiendo relacionarse en un formato entendible para el diseño de un modelo de predicción.
   2. Uno de los factores condicionantes en las recaídas es la severidad de la enfermedad. Cuantificar el dolor que siente un paciente no es tarea fácil. Este proyecto clasifica este factor en dos niveles, atendiendo a medidas objetivas que condicionan este sentimiento de dolor o gravedad. Este abordaje es similar a la clasificación del [dolor en la gripe estacional](https://www.cdc.gov/flu/about/classifies-flu-severity.htm) [9] que hace CDC (“Centers for Disease Control and Prevention”).

El conocimiento del momento de recaída de un paciente ayuda a obtener más información para gestionar los recursos clínicos, tanto personales como de infraestructura. Se percibe que hay mucha necesidad en el ámbito hospitalario para la toma de decisiones clínicas fiables y rápidas. A su vez, es útil contar con datos de muchos pacientes y de varios factores, ya que si las correlaciones se comparten a tiempo con la comunidad científica, se pueden alinear la toma de decisiones y detección temprana, enriqueciendo la calidad de vida de los pacientes. Se permite también la mejor adaptación de los tratamientos.

## 1.2 Objetivos del Trabajo

1.2.1 Objetivos generales:

1.2.1.1 Identificar factores que puedan interferir en las recaídas de hospitalizaciones.

1.2.1.2 Identificar si el diagnóstico y tratamiento de enfermedades son favorables.

1.2.2 Objetivos específicos:

1.2.2.1 Aplicar técnicas NLP en un contexto bioinformático trabajando con un dataset EHR.

1.2.2.2 Proponer algoritmos para el análisis de tiempos entre visitas basados en la severidad de la enfermedad y prescripciones médicas.

1.2.2.3 Analizar los historiales clínicos y el momento del alta de hospitalización.

1.2.2.4 Facilitar la toma de decisiones clínicas para la mejor gestión de recursos personales y de infraestructura.

## 1.3 Enfoque y método seguido

* 1. La base de este proyecto es la correcta elección del dataset con el que se va a trabajar. Se invertirá una gran parte del tiempo (30% del tiempo total) en la investigación de datasets, evaluación y acceso.

Se evaluaron varias posibilidades, tanto datasets reales como sintéticos.

* Dataset real. Es un dataset generado por una fuente fiable, que garantiza la anonimización de los pacientes, y se pueden extraer conclusiones biomédicas consistentes. En este caso podemos centrarnos en una o múltiples enfermedades.

Este dataset puede comprometer la seguridad y privacidad de los individuos, y por esto, el acceso está restringido.

* Dataset sintético. Es un dataset creado artificialmente, por lo que no se pueden generar evidencias de que las conclusiones biomédicas extrapoladas a partir de los datos sean veraces. Es por esto que el dataset sintético carece de interés. En este caso tendríamos que centrarnos en un pipeline genérico y no en una enfermedad concreta ya que no se podrían extrapolar conclusiones biomédicas.

La metodología seguida una vez conseguido el acceso al dataset real ha sido:

* Configuración de los datos para trabajar en la nube.
* Adaptación del procesamiento y memoria de la instancia en la nube.
* Estudio del tipo de datos del dataset.
* Identificación de las columnas relevantes.
* Estudio longitudinal de la población.
* Categorización de datos.
* Limpieza de datos: eliminar valores nulos, eliminar caracteres especiales…
* Estudio de outliers.
* Agrupación por hospitalizaciones.
* Simplificación de códigos de diagnóstico.
* Simplificación de prescripciones médicas.
* Categorización de la severidad.
* Aplicación de etiquetas en las notas de los doctores, de acuerdo a corpus biomédicos previamente entrenados.
* Filtro por una enfermedad en particular: Artritis Reumatoide (AR) y por las enfermedades más comunes.
* Estudio de los tiempos entre visitas.
* Relación de los eventos identificados en la vida del paciente (número de ingresos, número de hospitalizaciones en UCI, severidad de la enfermedad, visitas al hospital, prescripciones médicas) con el diagnóstico de enfermedades.
* Modelo de predicción de tiempos de rehospitalización para nuevos pacientes.

La agrupación de los datos por hospitalizaciones ha sido clave para el análisis. Puesto que no estamos realizando diagnósticos de enfermedades, podemos asumir que las hospitalizaciones de cada paciente son independientes, lo que permite incrementar la cantidad de datos disponibles.

Se ha decidido trabajar con Python, en un servidor “Jupyter notebook” implementado en una instancia ec2 de Amazon Web Services (AWS), con estas especificaciones: t2.xlarge con 200GB.

El acceso a los datos fuente se ve restringido debido a su contenido sensible. Es por ello que se hace uso de “pickles” para guardar el contenido de las tablas más importantes y se pueda seguir la ejecución.

## 1.4 Planificación del Trabajo

La planificación inicial ha sido ligeramente modificada durante el mes de Diciembre, ya que surgieron un par de tareas con las que no se contaba inicialmente:

* One-hot encoding
* Word2Vec

Esto ha retrasado la entrega del último Checkpoint, pero no la entrega de la memoria y resultados finales, puesto que en la planificación inicial se contaba con tiempo suficiente de margen.

* 1. ****

## 1.5 Breve sumario de productos obtenidos

* 1. Los entregables de este proyecto son:
* Esquema de las tablas.
* “Master.ipynb” jupyter notebook.
* “Pickles.ipynb” jupyter notebook

Ambos notebooks describen los mismos pasos, siendo “pickles.ipynb” la versión simplificada. Esta cuenta con la exportación de datos en diferentes puntos del notebook “Master.ipynb” para seguir mejor la ejecución.

## 1.6 Breve descripción de los otros capítulos de la memoria

* Arquitectura de tablas

Descripción de las tablas usadas del total disponible en la base de datos MIMIC-III.

Descripción de cómo se cargan estas tablas y a qué *dataframes* dan lugar en este proyecto.

* Conexión con la instancia ec2 de AWS

Descripción de las especificaciones de la máquina virtual, requisitos de seguridad y pasos de conexión.

* Estudio longitudinal de la población

Análisis de los pacientes disponibles y sesgado a través de la ejecución del proyecto por razones convenientes.

* Severidad

Descripción de la variable severidad y de su creación.

* Diagnósticos

Análisis de los diagnósticos más comunes y criterios de elección.

* Prescripciones

Análisis de las prescripciones más comunes y criterios de elección.

* Notas del doctor

Análisis de las notas de texto libre, criterios de extracción del historial clínico y del momento del alta del paciente.

Descripción de la transformación de texto a vectores mediante la técnica Word2Vec.

* Tiempo entre rehospitalizaciones

Creación de la variable objetivo para el diseño del modelo.

* Organización del dataset

Descripción de la técnica one-hot encoding simplificado.

* Modelos

Parámetros de diseño y comparación de modelos. Análisis según el parámetro “accuracy”.

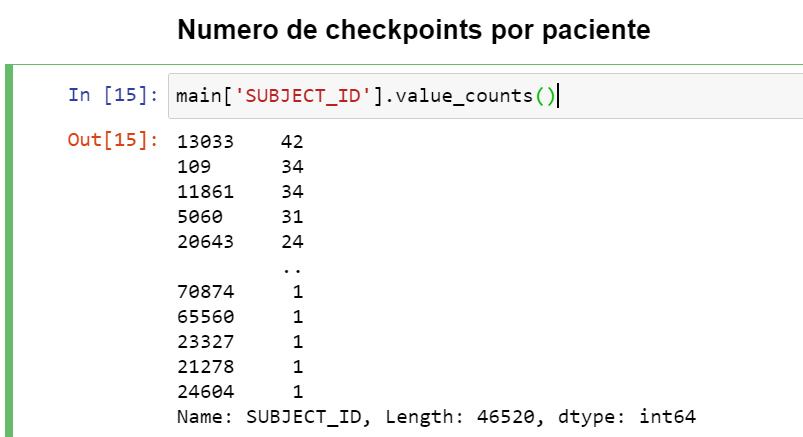
# 2. Desarrollo

### 2.1 Arquitectura de tablas

Estas son las tablas útiles. Se pueden consultar [aquí](https://mimic.physionet.org/about/mimic/) [10] todas las tablas disponibles en MIMIC-III.

* Admissions

Da información acerca de las admisiones de los pacientes en el hospital. Dado que cada visita al hospital es única, se cuenta con un identificador único ‘HADM\_ID’. Un paciente con ID ‘SUBJECT\_ID’ puede tener relacionadas varias visitas al hospital. Llamaremos **checkpoint por paciente** al número de visitas al hospital por paciente. Se observa que el paciente con más número de visitas es el paciente con ID=13033, que tiene 42 visitas.



Las columnas útiles de esta tabla son:

* + Subject\_ID: identificador único del paciente.
  + Hadm\_ID: identificador único de visita al hospital.
  + AdmitTime fecha de admisión.
  + DischTime: fecha de alta.

Al organizar la tabla master por hospitalizaciones, encontraremos la variable ‘checkpoints’ como una suma acumulativa de los ingresos agrupados por paciente.

* Icustays

Contiene todas las hospitalizaciones en la UCI por paciente y por ingreso. Puede ocurrir que en un mismo ingreso HADM\_ID haya varias hospitalizaciones en UCI, varias ICUSTAY\_IDs (ejemplo, paciente Subject\_ID=109).

Las columnas útiles de esta tabla son:

* + Subject\_ID: identificador único del paciente.
  + Hadm\_ID: identificador único de visita al hospital.
  + ICUSTAY\_ID: identificador único de hospitalización en la UCI.
* Prescriptions: contiene las prescripciones médicas.

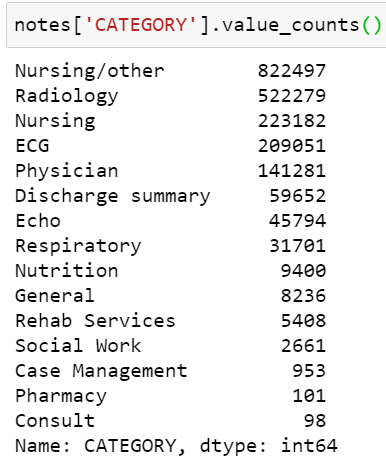
Las columnas útiles de esta tabla son:

* + Subject\_ID: identificador único del paciente.
  + Hadm\_ID: identificador único de visita al hospital.
  + DRUG: medicamento prescrito. Puede haber varios por hospitalización.
* Notes

Contiene todas las notas del doctor por paciente y visita al hospital.

Las columnas útiles de esta tabla son:

* Subject\_ID: identificador único del paciente.
* Hadm\_ID: identificador único de visita al hospital.
* Category: define el tipo de nota. Estos son los posibles valores de categorías disponibles:



Se han analizado las notas de categoría ‘Discharge Summary’ ya que son las relativas al alta del paciente y relacionadas con las preguntas a responder.

* Description: contiene la descripción de la categoría de la nota. Puede ser ‘Report’, ‘Anexo’, …
* IsError: indica que un científico ha identificado una nota como errónea (1=Error). Se filtra este tipo de notas porque no son de interés. Se borran 886 notas.
* Text: contiene el texto de la nota. Se observan varias secciones dentro de esta columna.

'service','allergies','attending','surgical','history\_current','history\_past','history\_social','history\_family','phy\_exam','neuro\_exam','results','hospital\_course','medicatons\_admission','medications\_discharge','disch\_disp','disch\_dia','disch\_cond','disch\_instr','Followup','End'

Las más relevantes son:

* History of present illness.
* Todas las relativas al Discharge.

Se ha aplicado “Data cleaning” para eliminar los retornos de carro, los símbolos raros (ejemplo [,],\*\*...), convertir a minúsculas, separar secciones de texto relevantes, eliminar filas con celdas vacías.

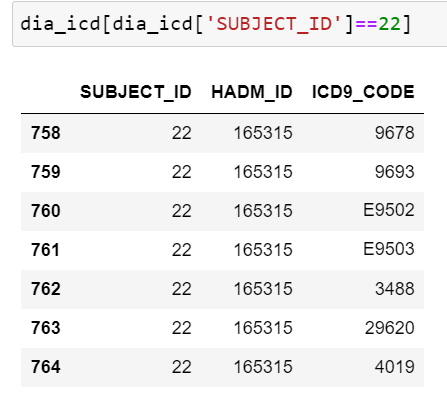
* Diagnoses\_ICD

Contiene los diagnósticos ICD9\_CODE por paciente y por visita al hospital.

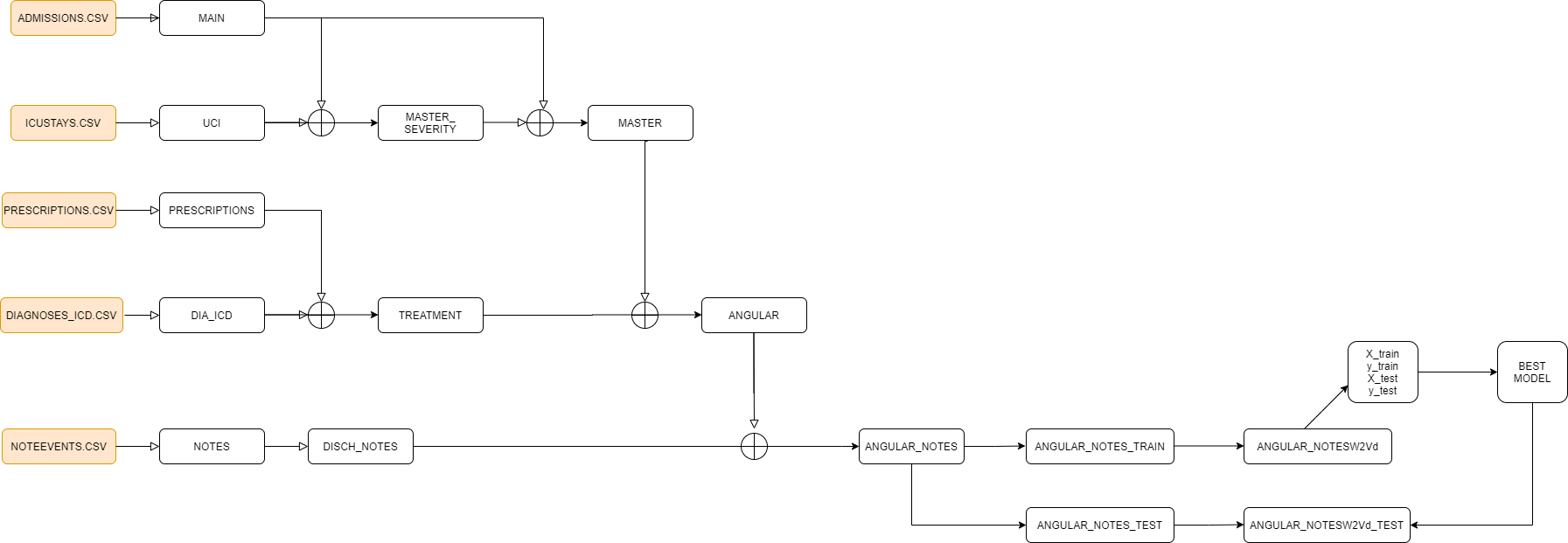
Las columnas útiles son:

* Subject\_ID
* Hadm\_ID
* ICD9\_Code

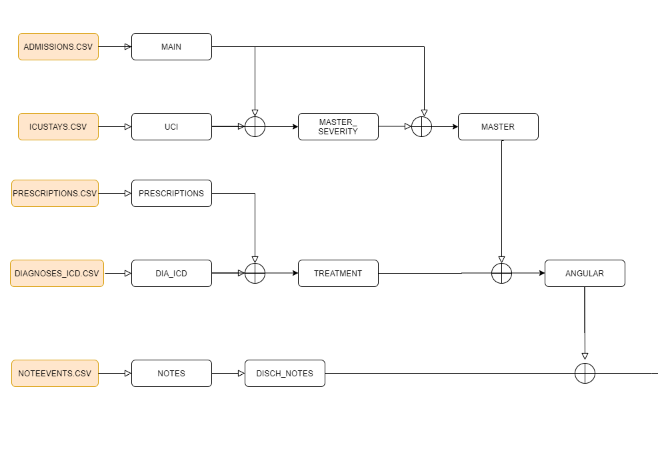
Un mismo paciente puede estar diagnosticado por varios códigos en la misma visita al hospital. Ejemplo: paciente con ID=22:

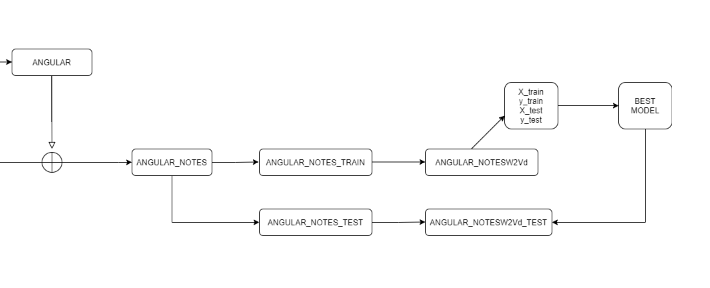


Este es el diagrama completo de la arquitectura de tablas.



Se divide en dos el gráfico para favorecer la visualización:



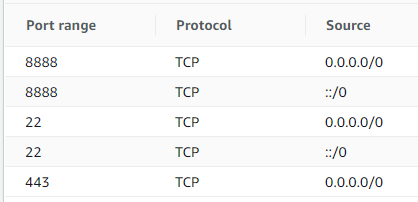


### 2.2 Conexión con la instancia ec2 de AWS

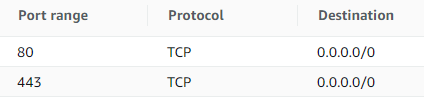
La instancia ec2 de AWS es del tipo **t2.xlarge** con **200GB**.

A la hora de configurar la conexión es necesario prestar atención a las reglas de seguridad.

Estas son las reglas “Inbound”:



Estas son las reglas “Outbound”:



A través de la terminal se conecta con la instancia con el siguiente comando:

ssh -i .pem\_path ec2-user@”PublicDNS”,

donde .pem\_path es la ruta de acceso al fichero de configuración .pem, que contiene los credenciales de seguridad (claves pública y privada). En el anexo se incluye cómo realizar esta configuración [11].

Una vez realizada la conexión, abrimos un servidor “Jupyter notebook” en el puerto 8888, al cual accederemos a través de ”PublicDNS”:8888, en una pestaña del navegador. Desde aquí tenemos acceso a los datos y notebooks de Python.

### 

### 2.3 Estudio longitudinal de la población

La distribución de los tiempos de admisión y de alta están descritos por 58976 filas (58976 hospitalizaciones de 46520 pacientes) en la tabla de admisiones.

No todas ellas contienen notas del doctor, resultando en 39641 filas (39166 hospitalizaciones de 30423 pacientes) para el diseño del modelo.

Los modelos son entrenados con la técnica cross-validation. Los datos etiquetados conforman el 23.71% de los datos disponibles (10112 filas, 9218 hospitalizaciones de 5676), usando el 80% para su entrenamiento (8089 filas, 7499 hospitalizaciones de 4940 pacientes).

Puede no coincidir el número único de hospitalizaciones con el número total de filas porque en una misma hospitalización pueden existir diferentes ingresos en la UCI.

### 2.4 Severidad

Se ha creado la variable “Severity” que indica el grado de dolor que experimenta un paciente entre las dos opciones disponibles: **Low** or **High**. La estrategia seguida se basa en la que se usa para [determinar la severidad en la gripe](https://www.cdc.gov/flu/about/classifies-flu-severity.htm), pero adaptada a nuestros datos. Estas son las variables a tener en cuenta en el cálculo de la severidad:

* Número de veces que el paciente ha visitado al hospital (número de checkpoints por paciente)
* Número de hospitalizaciones en la UCI
* Si el paciente sigue vivo o ha fallecido

Este es el criterio de decisión:

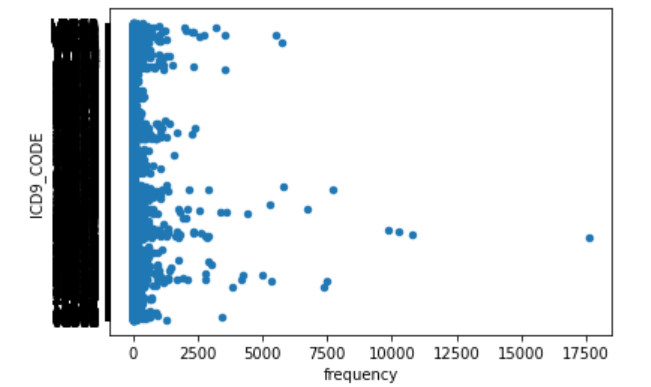
* High: si el paciente ha fallecido o si aún estando vivo ha estado hospitalizado en la UCI 3 o más veces.
* Low: si el paciente sigue vivo y o no ha estado en la UCI o ha estado menos de 3 veces.

### 2.5 Diagnósticos

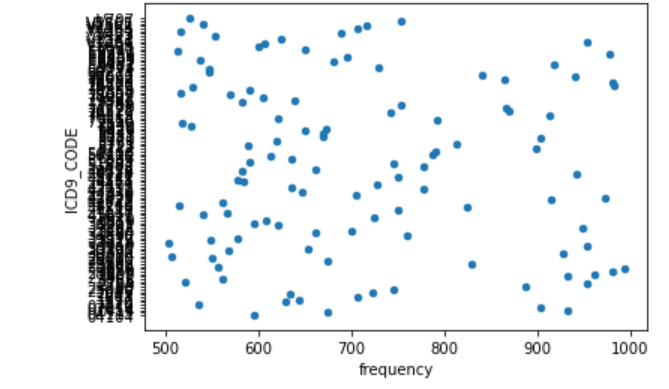
El número de diagnósticos posibles es de 6984, y se han elegido 118 (117 + ‘Others’). Se han perdido el 29.5% de pacientes con una enfermedad concreta (distinta de ‘Others’) pero se ha reducido en un 98.32% el número de enfermedades posibles.

Se ha comprobado que todos los pacientes tengan al menos un diagnóstico.

A continuación se presentan los diferentes gráficos que han servido para determinar el umbral:



Se ha eliminado el outlier y la frecuencia menor de 500. Se ha elegido 500 porque se quería asegurar la aparición de la enfermedad Artritis Reumatoide, cuyo punto de corte es frecuencia=518.

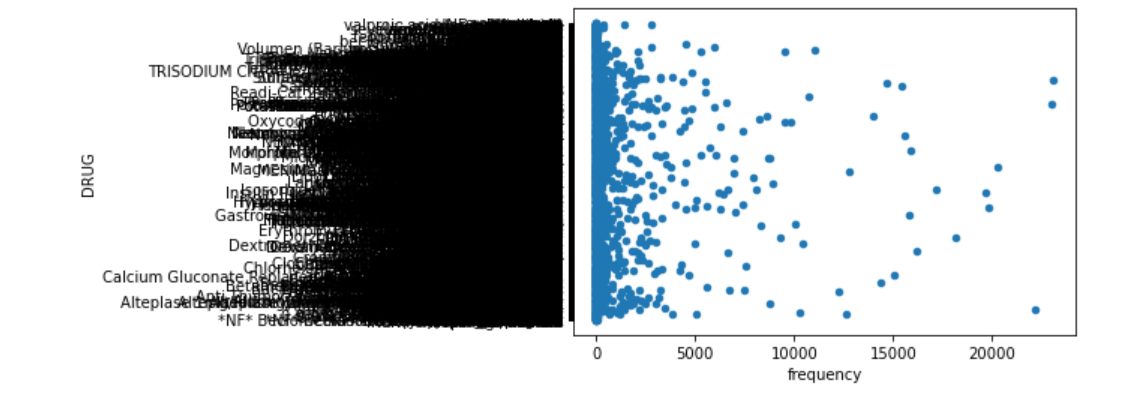


### 2.6 Prescripciones

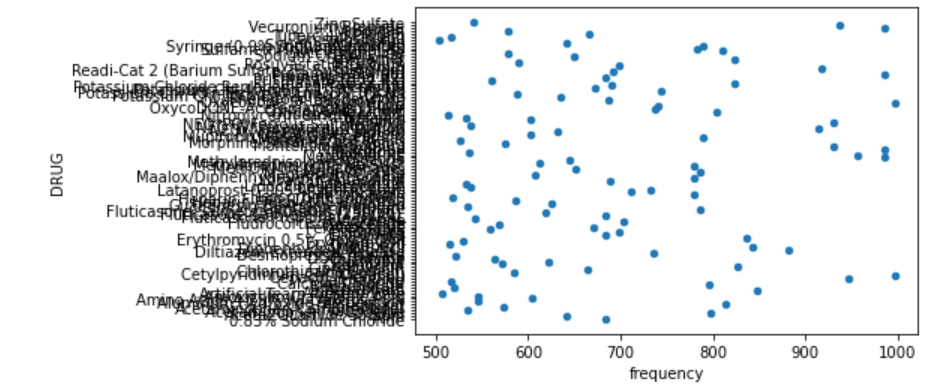
El número de prescripciones posibles es de 4220, y se han elegido 104 (103 + ‘Others’). Por lo tanto, el 97.5% de las prescripciones han sido categorizadas como ‘Others’.

Se ha comprobado que todos los pacientes tengan al menos una prescripción médica.

A continuación se presentan los diferentes gráficos que han servido para determinar el umbral:



Se ha eliminado la frecuencia menor de 500. Se establece el mismo criterio que en el caso de los diagnósticos.



### 2.7 Notas del doctor

Las notas escritas por el doctor es texto libre, que se ha de representar a través de números para poder ser entendido por el modelo en cuestión. Se hace uso de la técnica Word2Vec (W2V), que a partir de un corpus biomédico pre-entrenado (sciscpacy, ‘en\_core\_sci\_sm’), cada entrada de texto se traduce en un vector de 96 elementos, según la distancia vectorial con el corpus. De tal forma que cuanta menos distancia exista entre dos palabras, más probabilidad existe de que aparezcan juntas en el mismo contexto.

Contamos con dos columnas de texto libre: el historial clínico y el momento del alta del paciente. Entre ambas se cuenta con 192 (96\*2) columnas.

Todas las notas han sido filtradas por un tipo en concreto ‘Discharge summaries’ y porque no sean corruptas (filtrando por la variable ‘ISERROR’).

### 2.8 Tiempo entre rehospitalizaciones

Se ha creado la variable ‘tiempo entre re-hospitalizaciones’ que servirá de etiqueta para el conjunto de entrenamiento. Ha sido calculada de la siguiente manera:

* Identificar el usuario
* Obtener la fecha de alta de la hospitalización actual
* Buscar el resto de hospitalizaciones
* Descartar todas las hospitalizaciones previas a la actual
* Seleccionar la más cercana a la hospitalización actual.
* Computar la diferencia entre el alta de la hospitalización actual y el futuro ingreso.

Se hace notar que a aquellas hospitalizaciones con número máximo de ingresos no les aplica esta variable ya que no ha habido re-hospitalización. Precisamente estos pacientes serán los que formen parte del conjunto de datos de nuevas predicciones. Por lo tanto, si un paciente tiene 34 checkpoints, contará con 33 etiquetas. Se computa a 0 el tiempo de re-hospitalización del máximo checkpoint (el de 34).

### 2.9 Organización del dataset

El dataset se organiza por hospitalizaciones en vez de por pacientes. De tal forma que si un paciente tiene varios ingresos, estos se tratan de forma independiente. Esta decisión no tiene impacto en el resultado porque no se está haciendo análisis de diagnóstico.

En cada fila hay una única hospitalización representada. De tal forma que como contamos con varios diagnósticos por hospitalización, y varias prescripciones, la tabla debe convertirse para que esos valores se transformen en columnas. Para ello, se usa la técnica one-hot encoding. Esta consiste en representar con ‘0’ y ‘1’ la tabla según una hospitalización cuente (‘1’) o no cuente (‘0’) con un diagnóstico o prescripción determinados.

A su vez, se ha optado por simplificar la tabla “master”, quedando finalmente con 423 columnas. La forma de simplificar se ha descrito ya en apartados anteriores (2.5 y 2.6).

Todas las hospitalizaciones cuentan con una fecha de admisión y una fecha de alta. Por lo que no existe paciente ingresado en progreso. Puede ser que la razón por la que los pacientes no estén en el hospital sea el fallecimiento. Para estos casos, no hay hospitalización posible. Esto ha sido tenido en cuenta, y este tipo de casos de uso forman parte del conjunto de datos sin etiquetar para hacer predicciones con ellos. No se han quitado del dataset por contar con más datos. Una posible mejora sería eliminarlos en una futura iteración.

### 2.10 Modelos

Los datos etiquetados se dividen en un 80% de entrenamiento y un 20% de test.

Con el uso de ese 80% se construyen varios modelos y se elige el mejor (testeando con el 20% resultante), el cual se entrena de nuevo con el 100% de los datos etiquetados.

Este modelo pasa entonces a evaluar los datos no etiquetados. Al no poseer etiquetas, no es posible conocer el ‘accuracy’ sobre estos datos.

Al hacer modelos de clasificación, la variable objetivo tiene que ser de tipo ‘int’, por lo que se debe redondear la variable ‘tiempo entre rehospitalizaciones’, dando por válido el número 0 como predicción. Este 0 de predicción quiere decir que la siguiente hospitalización se producirá en menos de 24 horas.

Aunque sea obvio, cabe decir que la variable objetivo debe quitarse del dataset de entrenamiento. Si no fuera así, estaríamos haciendo trampas al modelo y obtendríamos ‘accuracy’ mucho mejores pero no reales.

Los modelos comparados son:

* Random Forest
* Naive Bayes
* Support Vector Machine
* Árbol de decisión

El modelo que aporta mejor precisión es el árbol de decisión. Sin embargo, este valor es muy bajo.

Como siguientes iteraciones, se ajustarían los hiperparámetros, se podría aumentar los datos de entrenamiento incluyendo otro tipo de notas, se pueden tratar los outliers o los valores nulos, o se podrían usar métodos de ensamblado para combinar predicciones de múltiples modelos de una sola vez. O incluso se podría abordar la selección de variables, ya que quizá no todas ellas aporten valor en la predicción.

# 3. Conclusiones

1. Mi conclusión más certera es acerca del acceso a datos reales. No es sencillo disponer de datos de tu interés, y de existir, cuesta mucho tiempo el conseguir el acceso. Es entendible puesto que hay información sensible, pero a su vez es un cuello de botella tremendo, que puede retrasar las investigaciones, o incluso desechar muchas de ellas. Entiendo que en el ámbito académico será diferente del ámbito profesional científico, o eso espero.

Inicialmente se sopesó apoyar fuertemente la tecnología Natural Language Processing (NLP) para hacer análisis del texto libre y extraer correlaciones de eventos. Incluso se contaba con información acerca de las pruebas de laboratorio. Finalmente, la falta de tiempo no ha dejado espacio para este aprendizaje. Sin embargo, he aprendido muchos cerca de EHR y su tratamiento, nuevas técnicas de procesamiento y cómo tratar datasets de grandes dimensiones.

Esperaba haber invertido más tiempo en el diseño de los modelos, lo cual tampoco ha sido posible, renunciando al manejo de los hiperparámetros.

La planificación ha seguido su curso, pero en el sprint final han surgido cosas que inicialmente no se esperaba. Esto ha hecho tener que renunciar a refinar mejor los resultados.

La comunicación asíncrona con el tutor a través del foro no siempre es eficiente. No nos culpo como individuos sino al método de comunicación que podría estar mejor diseñado.

Recapitulando, nuevas iteraciones de este proyecto serían:

* Predecir correlación de eventos.
* Analizar una enfermedad concreta y ver sus patrones de rehospitalización.
* Ajuste de hiperparámetros para mejorar el ‘accuracy’ de los modelos de aprendizaje.
* Eliminar los pacientes fallecidos del conjunto de datos de nuevas predicciones..
* Aplicar NLP para enriquecer el análisis.

# 4. Glosario

*W2V:* Word2Vec

*EHR:* Electronic Health Records

*MIMIC-III:* Medical Information Mart for Intensive Care - III

*AWS:* Amazon Web Services

* 1. *CDC:* Centers for Disease Control and Prevention

*AR:* Artritis Reumatoide

*NLP:* Natural Language Processing

# 5. Bibliografía

[1] Bahrami, M. (2016, April 19). *A dynamic cloud computing platform for eHealth systems.* Retrieved October 11, 2020, from <https://ieeexplore.ieee.org/document/7454539>

[2] Mayampurath, A. (2018). *Statistical Modeling of Clinical Data*. Retrieved October 11, 2020, from <https://cri.uchicago.edu/wp-content/uploads/2018/02/CRI_StatisticalModeling_Methods.pdf>

[3] Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L., & Mark, R. (2020, August 13). MIMIC-IV. Retrieved October 11, 2020, from <https://physionet.org/content/mimiciv/0.4/>

[4] Johnson, A., Pollard, T., Shen, L., Lehman, L., Feng, M., Ghassemi, M., . . . Mark, R. (2016, May 24). *MIMIC-III, a freely accessible critical care database*. Retrieved October 11, 2020, from <https://www.nature.com/articles/sdata201635>

[5] Long, A. (2019, July 09). *Introduction to Clinical Natural Language Processing: Predicting Hospital Readmission with Discharge Summaries.* Retrieved October 11, 2020, from <https://towardsdatascience.com/introduction-to-clinical-natural-language-processing-predicting-hospital-readmission-with-1736d52bc709>

[6] Medical School, H. (n.d.). *N2c2 NLP Research Data Sets.* Retrieved October 11, 2020, from <https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/>

[7] A. (n.d.). *SpaCy models for biomedical text processing*. Retrieved October 11, 2020, from <https://allenai.github.io/scispacy/>

[8] A. (n.d.). *MedaCy Documentation.* Retrieved October 11, 2020, from <https://medacy.readthedocs.io/en/latest/>

[9] C. (2018, September 14). How CDC Classifies Flu Severity. Retrieved December 31, 2020, from https://www.cdc.gov/flu/about/classifies-flu-severity.htm

[10] MIMIC III Critical Care Database. (n.d.). Retrieved December 31, 2020, from <https://mimic.physionet.org/about/mimic/>

[11] A. (n.d.). Amazon EC2 key pairs and Linux instances. Retrieved from https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html

# 6. Anexos

### 6.1 Amazon EC2 key pairs and Linux instances [11]

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance. Amazon EC2 stores the public key, and you store the private key. You use the private key, instead of a password, to securely access your instances. Anyone who possesses your private keys can connect to your instances, so it's important that you store your private keys in a secure place.

When you launch an instance, you are [prompted for a key pair](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launching-instance.html#step-7-review-instance-launch). If you plan to connect to the instance using SSH, you must specify a key pair. You can choose an existing key pair or create a new one. When your instance boots for the first time, the content of the public key that you specified at launch is placed on your Linux instance in an entry within ~/.ssh/authorized\_keys. When you connect to your Linux instance using SSH, to log in you must specify the private key that corresponds to the public key content. For more information about connecting to your instance, see [Connect to your Linux instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html). For more information about key pairs and Windows instances, see [Amazon EC2 key pairs and Windows instances](https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2-key-pairs.html) in the *Amazon EC2 User Guide for Windows Instances*

Because Amazon EC2 doesn't keep a copy of your private key, there is no way to recover a private key if you lose it. However, there can still be a way to connect to instances for which you've lost the private key. For more information, see [Connecting to your Linux instance if you lose your private key](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/replacing-lost-key-pair.html).

The keys that Amazon EC2 uses are 2048-bit SSH-2 RSA keys. You can have up to 5,000 key pairs per Region.

**Contents**

* [Creating or importing a key pair](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#prepare-key-pair)
* [Tagging a key pair](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#tag-key-pair)
* [Retrieving the public key for your key pair](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#retrieving-the-public-key)
* [Retrieving the public key for your key pair through instance metadata](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#retrieving-the-public-key-instance)
* [Locating the public key on an instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#locate-public-key-on-instance)
* [Identifying the key pair that was specified at launch](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#identify-key-pair-specified-at-launch)
* [(Optional) Verifying your key pair's fingerprint](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#verify-key-pair-fingerprints)
* [Adding or replacing a key pair for your instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#replacing-key-pair)
* [Connecting to your Linux instance if you lose your private key](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/replacing-lost-key-pair.html)
* [Deleting your key pair](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#delete-key-pair)

Creating or importing a key pair

You can use Amazon EC2 to create a new key pair, or you can import an existing key pair.

**Options**

* [Option 1: Create a key pair using Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#having-ec2-create-your-key-pair)
* [Option 2: Import your own public key to Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#how-to-generate-your-own-key-and-import-it-to-aws)

Option 1: Create a key pair using Amazon EC2

You can create a key pair using one of the following methods.

* [New console](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#new-console)
* [Old console](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#old-console)
* [AWS CLI](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#aws-cli)
* [PowerShell](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#powershell)

**To create your key pair**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
3. Choose **Create key pair**.
4. For **Name**, enter a descriptive name for the key pair. Amazon EC2 associates the public key with the name that you specify as the key name. A key name can include up to 255 ASCII characters. It can’t include leading or trailing spaces.
5. For **File format**, choose the format in which to save the private key. To save the private key in a format that can be used with OpenSSH, choose **pem**. To save the private key in a format that can be used with PuTTY, choose **ppk**.
6. Choose **Create key pair**.
7. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is determined by the file format you chose. Save the private key file in a safe place.  
   **Important**This is the only chance for you to save the private key file.
8. If you will use an SSH client on a macOS or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.
9. **chmod 400** *my-key-pair***.pem**
10. If you do not set these permissions, then you cannot connect to your instance using this key pair. For more information, see [Error: Unprotected private key file](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html#troubleshoot-unprotected-key).

Option 2: Import your own public key to Amazon EC2

Instead of using Amazon EC2 to create your key pair, you can create an RSA key pair using a third-party tool and then import the public key to Amazon EC2. For example, you can use **ssh-keygen** (a tool provided with the standard OpenSSH installation) to create a key pair. Alternatively, Java, Ruby, Python, and many other programming languages provide standard libraries that you can use to create an RSA key pair.

**Requirements**

* The following formats are supported:
  + OpenSSH public key format (the format in ~/.ssh/authorized\_keys). If you connect using SSH while using the EC2 Instance Connect API, the SSH2 format is also supported.
  + Base64 encoded DER format
  + SSH public key file format as specified in [RFC4716](http://tools.ietf.org/html/rfc4716)
  + SSH private key file format must be PEM (for example, use ssh-keygen -m PEM to convert the OpenSSH key into the PEM format)
* Create an RSA key. Amazon EC2 does not accept DSA keys.
* The supported lengths are 1024, 2048, and 4096. If you connect using SSH while using the EC2 Instance Connect API, the supported lengths are 2048 and 4096.

**To create a key pair using a third-party tool**

1. Generate a key pair with a third-party tool of your choice.
2. Save the public key to a local file. For example, ~/.ssh/my-key-pair.pub (Linux) or C:\keys\my-key-pair.pub (Windows). The file name extension for this file is not important.
3. Save the private key to a different local file that has the .pem extension. For example, ~/.ssh/my-key-pair.pem (Linux) or C:\keys\my-key-pair.pem (Windows). Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

After you have created the key pair, use one of the following methods to import your key pair to Amazon EC2.

* [New console](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#new-console)
* [Old console](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#old-console)
* [AWS CLI](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#aws-cli)
* [PowerShell](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#powershell)

**To import the public key**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Key Pairs**.
3. Choose **Import key pair**.
4. For **Name**, enter a descriptive name for the key pair. The name can include up to 255 ASCII characters. It can’t include leading or trailing spaces.
5. Either choose **Browse** to navigate to and select your public key, or paste the contents of your public key into the **Public key contents** field.
6. Choose **Import key pair**.
7. Verify that the key pair you imported appears in the list of key pairs.

Tagging a key pair

To help categorize and manage your existing key pairs, you can tag them with custom metadata. For more information about how tags work, see [Tagging your Amazon EC2 resources](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html).

You can view, add, and delete tags using the new console and the command line tools.

* [New console](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#new-console)
* [AWS CLI](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#aws-cli)
* [PowerShell](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#powershell)

**To view, add, or delete a tag for an existing key pair**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Key Pairs**.
3. Select a key pair, and then choose **Actions**, **Manage tags**.
4. The **Manage tags** page displays any tags that are assigned to the key pair.
   * To add a tag, choose **Add tag**, and then enter the tag key and value. You can add up to 50 tags per key pair. For more information, see [Tag restrictions](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html#tag-restrictions).
   * To delete a tag, choose **Remove** next to the tag to delete.
5. Choose **Save**.

Retrieving the public key for your key pair

On your local Linux or macOS computer, you can use the **ssh-keygen** command to retrieve the public key for your key pair. Specify the path where you downloaded your private key (the .pem file).

**ssh-keygen -y -f /***path\_to\_key\_pair***/***my-key-pair.pem*

The command returns the public key, as shown in the following example.

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V

hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr

lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ

qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb

BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE

If the command fails, run the following command to ensure that you've changed the permissions on your key pair file so that only you can view it.

**chmod 400** *my-key-pair***.pem**

Retrieving the public key for your key pair through instance metadata

The public key that you specified when you launched an instance is also available to you through its instance metadata. To view the public key that you specified when launching the instance, use the following command from your instance:

* [IMDSv2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#imdsv2)
* [IMDSv1](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#imdsv1)

[ec2-user ~]$ **TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \**

**&& curl -H "X-aws-ec2-metadata-token: $TOKEN" –v http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key**

The following is an example output.

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V

hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr

lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ

qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb

BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair

If you change the key pair that you use to connect to the instance, we don't update the instance metadata to show the new public key. Instead, the instance metadata continues to show the public key for the key pair that you specified when you launched the instance. For more information, see [Retrieving instance metadata](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html).

Alternatively, on a Linux instance, the public key content is placed in an entry within ~/.ssh/authorized\_keys. You can open this file in an editor. The following is an example entry for the key pair named **my-key-pair**. It consists of the public key followed by the name of the key pair.

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V

hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr

lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ

qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb

BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair

Locating the public key on an instance

When you launch an instance, you are [prompted for a key pair](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launching-instance.html#step-7-review-instance-launch). If you plan to connect to the instance using SSH, you must specify a key pair. You can choose an existing key pair or create a new one. When your instance boots for the first time, the content of the public key that you specified at launch is placed on your Linux instance in an entry within ~/.ssh/authorized\_keys.

**To locate the public key on an instance**

1. Connect to your instance. For more information, see [Connect to your Linux instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html).
2. In the terminal window, open the authorized\_keys file using your favorite text editor (such as **vim** or **nano**).
3. [ec2-user ~]$ **nano ~/.ssh/authorized\_keys**
4. The authorized\_keys file opens, displaying the public key, as shown in the following example.
5. ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6Vhz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXrlsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZqaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3RbBQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE

Identifying the key pair that was specified at launch

When you launch an instance, you are [prompted for a key pair](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launching-instance.html#step-7-review-instance-launch). If you plan to connect to the instance using SSH, you must specify a key pair.

* [New console](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#new-console)
* [Old console](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#old-console)

**To identify the key pair that was specified at launch**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. On the **Details** tab, under **Instance details**, the **Key pair name** field displays the name of the key pair that you specified when you launched the instance. The value of the **Key pair name** does not change even if you change the public key on the instance, or add key pairs.

(Optional) Verifying your key pair's fingerprint

On the **Key Pairs** page in the Amazon EC2 console, the **Fingerprint** column displays the fingerprints generated from your key pairs. AWS calculates the fingerprint differently depending on whether the key pair was generated by AWS or a third-party tool. If you created the key pair using AWS, the fingerprint is calculated using an SHA-1 hash function. If you created the key pair with a third-party tool and uploaded the public key to AWS, or if you generated a new public key from an existing AWS-created private key and uploaded it to AWS, the fingerprint is calculated using an MD5 hash function.

You can use the SSH2 fingerprint that's displayed on the **Key Pairs** page to verify that the private key you have on your local machine matches the public key stored in AWS. From the computer where you downloaded the private key file, generate an SSH2 fingerprint from the private key file. The output should match the fingerprint that's displayed in the console.

If you created your key pair using AWS, you can use the OpenSSL tools to generate a fingerprint as shown in the following example.

$ **openssl pkcs8 -in** *path\_to\_private\_key* **-inform PEM -outform DER -topk8 -nocrypt | openssl sha1 -c**

If you created a key pair using a third-party tool and uploaded the public key to AWS, you can use the OpenSSL tools to generate the fingerprint as shown in the following example.

$ **openssl rsa -in** *path\_to\_private\_key* **-pubout -outform DER | openssl md5 -c**

If you created an OpenSSH key pair using OpenSSH 7.8 or later and uploaded the public key to AWS, you can use **ssh-keygen** to generate the fingerprint as shown in the following example.

$ **ssh-keygen -ef** *path\_to\_private\_key* **-m PEM | openssl rsa -RSAPublicKey\_in -outform DER | openssl md5 -c**

Adding or replacing a key pair for your instance

You can change the key pair that is used to access the default system account of your instance. For example, if a user in your organization requires access to the system user account using a separate key pair, you can add that key pair to your instance. Or, if someone has a copy of the .pem file and you want to prevent them from connecting to your instance (for example, if they've left your organization), you can replace the key pair with a new one.

To add or replace a key pair, you must be able to connect to your instance. If you've lost your existing private key or you launched your instance without a key pair, you won't be able connect to your instance and therefore won't be able to add or replace a key pair. If you've lost your existing private key, you might be able to retrieve it. For more information, see [Connecting to your Linux instance if you lose your private key](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/replacing-lost-key-pair.html). If you launched your instance without a key pair, you won't be able to connect to the instance unless you chose an AMI that is configured to allow users another way to log in.

**Note**

These procedures are for modifying the key pair for the default user account, such as ec2-user. For more information about adding user accounts to your instance, see [Managing user accounts on your Amazon Linux instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-users.html).

**To add or replace a key pair**

1. Create a new key pair using [the Amazon EC2 console](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#having-ec2-create-your-key-pair) or a [third-party tool](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#how-to-generate-your-own-key-and-import-it-to-aws).
2. Retrieve the public key from your new key pair. For more information, see [Retrieving the public key for your key pair](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#retrieving-the-public-key).
3. Connect to your instance using your existing private key file.
4. Using a text editor of your choice, open the .ssh/authorized\_keys file on the instance. Paste the public key information from your new key pair underneath the existing public key information. Save the file.
5. Disconnect from your instance, and test that you can connect to your instance using the new private key file.
6. (Optional) If you're replacing an existing key pair, connect to your instance and delete the public key information for the original key pair from the .ssh/authorized\_keys file.

**Note**

If you're using an Auto Scaling group, ensure that the key pair you're replacing is not specified in your launch template or launch configuration. Amazon EC2 Auto Scaling launches a replacement instance if it detects an unhealthy instance; however, the instance launch fails if the key pair cannot be found.

Deleting your key pair

When you delete a key pair, you are only deleting the Amazon EC2 copy of the public key. Deleting a key pair doesn't affect the private key on your computer or the public key on any instances that already launched using that key pair. You can't launch a new instance using a deleted key pair, but you can continue to connect to any instances that you launched using a deleted key pair, as long as you still have the private key (.pem) file.

If you're using an Auto Scaling group (for example, in an Elastic Beanstalk environment), ensure that the key pair you're deleting is not specified in your launch configuration. Amazon EC2 Auto Scaling launches a replacement instance if it detects an unhealthy instance; however, the instance launch fails if the key pair cannot be found.

You can delete a key pair using one of the following methods.

* [New console](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#new-console)
* [Old console](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#old-console)
* [AWS CLI](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#aws-cli)
* [PowerShell](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#powershell)

**To delete your key pair**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Key Pairs**.
3. Select the key pair to delete and choose **Delete**.
4. In the confirmation field, enter Delete and then choose **Delete**.

If you create a Linux AMI from an instance, and then use the AMI to launch a new instance in a different Region or account, the new instance includes the public key from the original instance. This enables you to connect to the new instance using the same private key file as your original instance. You can remove this public key from your instance by removing its entry from the .ssh/authorized\_keys file using a text editor of your choice. For more information about managing users on your instance and providing remote access using a specific key pair, see [Managing user accounts on your Amazon Linux instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-users.html).