Análisis de Datos 2024

Practico 1: Obtención, exploración y pretratamiento de datos



**Integrantes**: Patrick Hobbins, Nicolas Silva, Nicolas Sere, Felipe Ham, Marcos Montenegro



***INDICE***

[1. Carga de Datos 2](#_Toc1413352759)

[2. Resumen de las Variables 2](#_Toc1642309793)

[3. Análisis Exploratorio de los Datos 2](#_Toc1165055184)

[4. Limpieza de Datos 6](#_Toc1254078371)

[5. Normalización de Datos 6](#_Toc331696553)

[6. Comparación de inconsistencias 6](#_Toc1190678980)

# 1. Carga de Datos

El primer paso fue cargar el dataset “permisos\_construccion.csv” para comenzar a trabajar con él. Este archivo tiene información detallada sobre los permisos de construcción en San Francisco (Estados Unidos), como números de permiso, tipos de permiso, fechas de creación, direcciones, entre otros. A partir de aquí, se comenzó la exploración y pretratamiento de los datos.

# 2. Resumen de las Variables

Cada columna en el dataset representa una variable que contiene información específica sobre los permisos de construcción.

Algunas de las variables importantes son:

* **Permit Number:** Número único de identificación del permiso.
* Street Number Suffix: los datos vacíos pueden ser porque no se conocen o no apliquen para esta dirección. Esto se dedujo porque hay muchos vacíos.
* Completed Date: los faltantes de esta columna deberían corresponder a las construcciones que aún no finalizaron.
* Structural Notification: hay una gran cantidad de datos faltantes, lo que significa que la gran mayoría de los permisos no tendrían que cumplir o tener en cuenta alguna notificación estructural.
* **Permit Type:** Código que indica el tipo de permiso emitido.
* **Permit Creation Date:** Fecha en que se creó el permiso.
* **Street Number, Street Name, Street Suffix:** Información relacionada con la dirección donde se realiza la construcción.
* **Completed Date:** Fecha en la que se completó el proyecto (si es aplicable).
* **Block, Lot:** Información sobre la parcela donde se lleva a cabo la construcción.

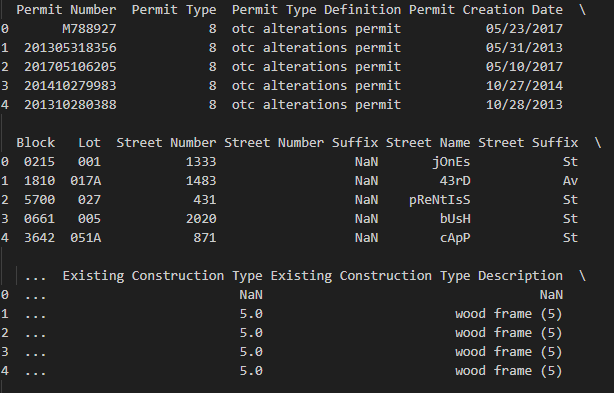
# 3. Análisis Exploratorio de los Datos

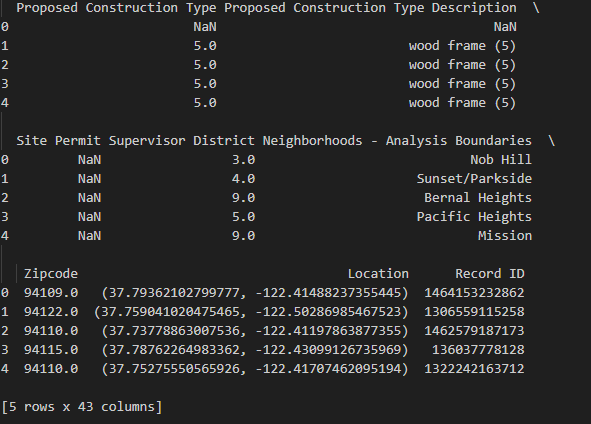
1. **Cantidad de filas y columnas**

El *dataset* está formado por 198910 filas y 43 columnas. Por lo tanto, hay casi 200000 registros divididos en 43 variables.

1. **Observación de las primeras 5 filas**

Luego se obtuvieron las primeras 5 filas para tener una primera vista de la información que tiene cada variable. Además, permite identificar problemas o inconsistencias desde el inicio.





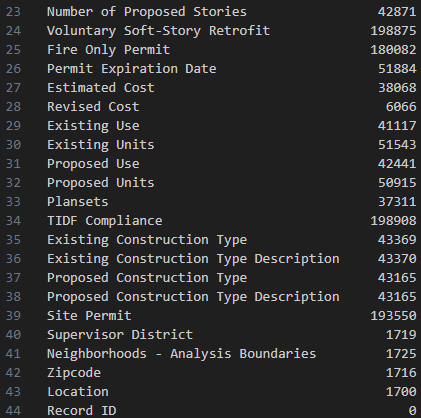
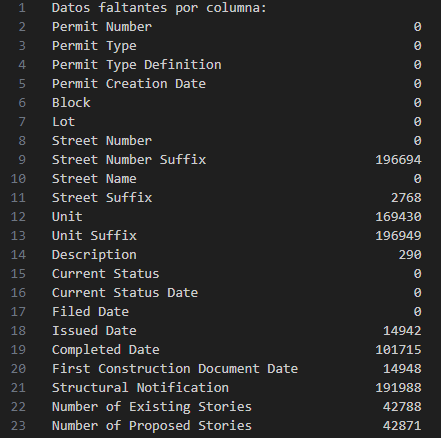
1. **Existencia de datos faltantes y duplicados.**

Durante la exploración, se identificaron datos faltantes en varias columnas:

Las columnas como ‘*Street Number Suffix’, ‘Unit’, y ‘Voluntary Soft-Story Retrofit’* presentan un alto porcentaje de datos faltantes, de los cuales se explicarán posibles motivos en la siguiente parte.

No se encontraron filas duplicadas. Por lo tanto, se puede deducir que cada registro es único.

La cantidad de datos faltantes de cada variable son los siguientes:



1. **Posibles motivos para los datos faltantes.**

Existen varios motivos por los cuales hay datos faltantes, así que a continuación se analizaran los motivos más importantes, en otras palabras, los motivos que abundan más y son más comunes.

Hay varios índices con el mismo motivo, y es que no aplican para este caso, como Street Number Suffix, Street Suffix, Unit Suffix, o Unit.

Mas adelante hay índices como Completed Date, que significa que los faltantes de esta columna deberían corresponder a las construcciones que aún no finalizaron. Luego se encuentran los índices como Structural Notification, que tienen una gran cantidad de datos faltantes, significando que la gran mayoría de los permisos no tendrían que cumplir o tener en cuenta alguna notificación estructural, o Voluntary Soft-Story Retrofit, que quiere decir que solo unos pocos permisos cuentan con la protección contra terremotos. También hay índices como Fire Only Permit, que quiere decir que son pocos los permisos relacionados a la prevención de incendios. Por último, se encuentra el TIDF Compliance, significando que hay un desconocimiento de este requerimiento legal.

Se explicó más detalladamente variable por variable, en la parte 3d del archivo ‘practico1.ipynb’, el motivo de los datos faltantes en cada una. Además, se analizó y tomó una decisión de qué hacer con cada columna en la parte 4.

1. **Evaluación de valores únicos en variables discretas**

Se evaluaron las variables categóricas. Son aquellas que pueden tomar un conjunto limitado de valores, como el ‘tipo de permiso’ o el ‘sufijo de la calle’. Esto ayuda a comprender la diversidad de valores que hay en cada variable y si algunos valores son más comunes que otros.

1. **Cuantificación de valores únicos y creación de histogramas**

Se cuantificaron los valores únicos de cada variable categórica y se generaron histogramas para poder ver la distribución de estos valores.

Los histogramas son útiles para identificar patrones en los datos, como si algunos tipos de permisos son mucho más comunes que otros.

1. **Evaluación de datos inconsistentes**

Revisamos las columnas de tipo objeto para asegurarnos de que no contengan números, y las columnas de tipo numérico para verificar que no incluyan texto. En un caso específico, encontramos una columna de tipo objeto que contenía un número escrito en palabras. Para corregir esto, convertimos la palabra al número correspondiente y cambiamos el tipo de la columna a numérico.

Otro ejemplo de datos inconsistentes que detectamos fue la mezcla de mayúsculas y minúsculas en palabras dentro de la columna "street name".

Realizaremos este tipo de análisis de forma más detallada para el resto de las columnas en el paso 4.

# 4. Limpieza de Datos

a) . Tomar una decisión respecto a los datos faltantes, duplicados, inconsistencias y cualquier inconveniente que encuentren con los datos.

Con respecto a los datos faltantes, en algunos casos decidimos eliminar las columnas porque cuando una columna tiene demasiados valores ausentes, mantenerla puede aportar poco valor, ya que la información disponible es limitada y podría no ser representativa, lo que podría introducir sesgos si se intenta analizar sobre los datos faltantes. Además, eliminar dichas columnas simplifica el modelo, permitiendo que este se enfoque en las características más relevantes y mejorando así su rendimiento y capacidad de generalización. También se evita la distorsión de resultados que podría ocurrir al rellenar una columna con muchos valores faltantes, especialmente si estos datos no son aleatorios y podrían seguir un patrón que sería difícil de modelar adecuadamente. Por último, al reducir la cantidad de columnas, se mejora la eficiencia computacional, optimizando el tiempo y los recursos necesarios para el análisis, especialmente en conjuntos de datos grandes. Entonces se eliminaron columnas como por ejemplo unit suffix o street number suffix.

En otros casos, lo que hicimos fue rellenar los datos faltantes dependiendo de varios factores, en el caso de los pisos, hay estructuras que no tienen pisos, como los puentes, en ese caso, se los puso un –1.

Con los datos duplicados se decidio ver cada columna por separado, ya que depende de varios factores a tomar en cuenta.

En los datos inconsistentes, como errores tipográficos o formatos mixtos (por ejemplo, direcciones en diferentes formatos), decidimos corregir estos valores para asegurar la uniformidad en el análisis, o sino en casos donde un valor numérico estaba escrito como texto, lo convertimos a su forma numérica y cambiamos el tipo de dato de la columna, ya que esto mejora la precisión y facilita cálculos posteriores, como ya comentado en la parte anterior (3g).

Acá esta explicado específicamente la razón de cada columna:

* Para la primera columna ‘Permit Number’ se dejó así ya que no había valores NaN. Los duplicados se dejan para analizar luego de analizar todas las demás variables para comparar toda la fila entera.
* Para la columna ‘Permit Type’ no parecen haber inconsistencias ya que efectivamente el tipo de permiso se representa numericamente y no faltan datos.
* En la columna 'Permit Type Definition', no había valores vacíos, pero sí se encontraron datos inconsistentes, como espacios adicionales o el símbolo '#' en algunos casos. Para los valores que terminaban con '#', se comparó la cantidad de veces que aparecían con y sin este símbolo. Dado que la mayoría de los valores no incluían el '#', se decidió eliminarlo. Además, se eliminaron los espacios en blanco al inicio y al final de los registros, así como los '#' presentes en algunos de ellos.
* Para los valores de la columna 'Permit Creation Date' no se tuvieron que agregar datos ya que no faltaba ninguno. Lo que sí se cambio fue el formato de algunas fechas, ya que algunas estaban con '-' y otras con '/'. Además, a los pocos valores que eran 'NaN' (5), se les colocó la fecha del primer documento de construcción para tener una fecha aproximada.
* Para los valores de la columna ‘Block’, no se cambió nada ya que no faltaban datos y no hay valores 'NaN', pareciendo estar sus valores correctos. Algunos tienen una letra, pero puede llegar a ser parte de su dirección.
* Al igual que la columna anterior 'Block', para 'Lot' no se cambió nada ya que no faltaban datos y no hay valores 'NaN', pareciendo estar sus valores correctos. Algunos tienen una letra, pero puede llegar a ser parte de su dirección.
* Al igual que las columnas anteriores 'Block' y 'Lot', para 'Street Number' no se cambió nada ya que no faltaban datos y no hay valores 'NaN', pareciendo estar sus valores correctos. El tipo de dato es int64 indicando que no hay letras en esta columna de numero de calle.
* Como hay un 98.89 % de datos faltantes del total de filas, se decidió borrar la columna 'Street Number Suffix' ya que no hay manera razonable de ponerle valores y además hay tantos valores faltantes que no aporta valor significativo al analisis.
* Para los valores de la columna 'Street Name' se les corrigió la inconsistencia de variar letras mayúsculas y minúsculas en cada palabra de los nombres de las calles. Se las puso todos con la primera letra mayúscula. Por otro lado, no tenía datos NaN o datos faltantes.
* A los valores NaN de 'Street Name Suffix', que representaban un 1,3% de todos los registros, se le les colocó 'St'. Se dibujó un histograma con la cantidad de veces que aparecía cada uno, y 'St' era significativamente el más predominante. Por lo tanto, al ser pocos valores comparado al total, se asumieron que eran calles (St de Street).
* En la columna ‘Unit’, como puede no aplicar a ciertas construcciones, cambiamos los valores nulos por –1
* En ‘Unit Suffix’ nos encontramos con más del 99% de los datos nulos, por lo que decidimos eliminar esta columna. Decidimos que no había forma de poner valores razonables a tanta can
* tidad de datos y que no sería significativo para el análisis.
* En la columna ‘Description’ decidimos sustituir los valores nulos por la frase “No description”, que nos pareció lo más razonable
* En ‘Current Status’ no notamos inconsistencias en los datos y ademas no hay valores nulos
* En ‘Current Status Date’ solamente cambiamos el formato de la fecha a dd/mm/aaaa ya que no encontramos valores nulos
* En ‘Filed Date’ realizamos lo mismo que en ‘Current Status Date’
* En ‘Issued Date’ decidimos cambiar los valores nulos por la fecha mínima encontrada en la columna, ya que creímos que era la decisión más acertada. También cambiamos el formato.
* En ‘Completed Date’ cambiamos el formato y sustituimos los valores nulos (tiene sentido que los haya ya que no todas las construcciones van a estar terminadas) por una fecha específica que sabremos qué significa (01/01/1900).
* En ‘First Construction Document Date’ realizamos exactamente lo mismo que en ‘Completed Date’
* En la columna ‘Structural Notification’ decidimos sustituir los valores por booleanos; los vacíos por False, ya que no tendrían una notificación, y los ‘Y’ por True.
* Con la columna ‘Number of Existing Stories’, tomamos los valores NaN y los cambiamos por –1, buscando que solo hubiera valores numéricos y que en algunos casos ‘Number of Existing Stories’ no aplicaba a algunos permisos.
* ‘Number of Proposed Stories’ era una columna donde se colocaban los pisos propuestos para el permiso de construcción, en caso de NaN lo remplazamos por los valores de Number Existing Stories, como nos aseguramos de que en la parte anterior los NaN fueran vistos como –1, nos aseguramos de que no quedaran en ‘Number of Proposed Stories’.
* En ‘Voluntary Soft-Story Retrofit’ habian valores “Y” y otros valores que eran NaN, decidimos tomar los “Y” y cambiarlos por True, por otro lado, tomamos los valores NaN como False, la columna paso a ser de tipo bool.
* ‘Fire Only Permit’ fue un caso similar, optamos por cambiar “Y” por True y los NaN por False.
* En el caso de ‘Permit Expiration Date’, tomamos los valores NaN y los cambiamos por la fecha maxima, de manera que en caso de no tener la información poder tener tiempo para formalizar la renovación.
* Para la columna de ‘Estimated Cost’ tomamos la media de los datos y reemplazamos los NaN con ello.
* Referido a ‘Revised Cost’, optamos por tomar los datos de ‘Estimated Cost’ y reemplazar con esos datos los NaN.
* Con ‘Existing Use’ tomamos los valores unicos, buscamos y reemplazamos, esto nos ayudara a posteriori con el ‘Proposed Use’, por otro lado, tomamos los valores NaN y los reemplazamos por “Desconocido”.
* ‘Proposed Use’ son los usos propuestos, en caso de NaN optamos por aplicar los valores de la columna ‘Existing Use’, de esa manera podemos asegurar una mayor correctitud.
* Para la columna de ‘Existing Units’, tomamos los NaN y los cambiamos por la media. Esto con el propósito de no tener una gran desviación en caso de querer generar un modelo de *Machine Learning.*
* ‘Proposed Units’ tuvo el mismo tratamiento que proposed Use, para los NaN tomamos los datos de la columna anterior y remplazamos. Nos aseguramos de que ningún dato sea negativo.
* Para los datos faltantes de la columna 'TIDF Compliance', se rellenaron los valores NaN con False. Esto se decidió para uniformizar los datos. Además,se buscaron duplicados, pero se excluyeron los valores NaN para garantizar que solo se consideraran valores significativos. Los duplicados encontrados se imprimieron y se ordenaron para facilitar su revisión.Por otro lado, se corrigieron datos inconsistentes, por ejemplo, los valores 'P' se reemplazaron con 'Y', y luego 'Y' se transformó en True, mientras que otros valores se cambiaron a False. Este enfoque asegura que los datos sean consistentes y que la columna sea de tipo booleano, lo cual támbien se hizo para asegurar consistencia y ahorrar espacio, así como para describir mejor los datos. El ahorro de memoria es significativo: mientras que un booleano se puede almacenar en un bit, tipicamente un string requiere al menos cincuenta bytes, es decir 400 veces más memoria por fila en dicha columna.

- 'Existing Construction Type' y 'Existing Construction Type Description' se manejaron juntas por su grado de cercanía y se realizo lo siguiente: búsqueda de valores nulos, para identificar registros con valores nulos en ambas columnas. Luego, se comparó si había alguna nula en una columna y en otra no, lo cual es una incostintencia: se encontró una columna de este tipo, que era además la unica con número -99999. Sin embargo, había cuatro columnas con el número 99999 y la frase “Wood Frame” con lo cual se cambio esta los valores de la fila incosistente a estos valores. Por último, relleno de nulos: reemplazó valores nulos en 'Existing Construction Type' con -1 y en 'Existing Construction Type Description' con "NON AVAILABLE" para estandarizar datos faltantes.

- En “Proposed Construction Type” y “ Proposed Construction Type Description” se procedió de la misma manera, no encontrando filas con valores nulos en una columna y no en la otra; por otro lado, se reemplazó los valores nulos siguiendo el mismo criterio que anteriormente.

- En “Site Permit” se preocedió de modo de cambiar los valores Nan por False, y los Y por True, procediendo luego a hacer el tipo de columna booleana por los beneficios mencionados anteriormente. No se encontraron datos que no fueran “Y” o Nan.

- En “Supervisor District” se encontraron valores nulos, y valores escritos como “quince” en lugar de 15. Por lo tanto, se cambian los números escritos como texto en la columna a sus equivalentes numéricos, se cambia la columna a tipo numérico, y se reemplazan los valores nulos con -1.

- En “Neighborhoods - Analysis Boundaries” se encontró como error valores nulos, que fueron sustituidos por “Non Avaivable”

- En “Zipcode” y “Location” se procedió de la misma manera.

Por último, se borraron todos los espacios al principio y al final de todas las celdas porque alguna había faltado.

Luego de la limpieza de datos columna por columna verificamos si verdaderamente no existían duplicados en el dataframe y nos encontramos con que había 10 filas duplicadas. Por lo tanto, eliminamos las que se repetían.

# 5. Normalización de Datos

a) La normalización es un proceso que ajusta los valores de las variables numéricas para que todos estén dentro de un rango específico, como [0, 1] o [-1, 1]. Esto es importante porque cuando las variables tienen escalas diferentes, algunas pueden tener más influencia en el análisis que otras. Normalizar ayuda a que todas las variables tengan el mismo impacto en el análisis.

Con respecto al código para la normalización, se seleccionaron las variables numéricas. Se sacaron las que no debían ser normalizadas como ‘Zipcode’ o ‘Record ID’ ya que no tenía sentido cambiar sus valores. Luego se identificaron las columnas continuas que deben ser normalizadas. Por último, se normalizaron estas columnas y se mostraron las primeras filas para ver los resultados.

b) A continuación se analizarán las ventajas de la normalización de datos y algunos de sus posibles usos.

**Ventajas de la normalización**

Primero, la normalización es útil porque iguala la influencia de las variables en modelos estadísticos y de machine learning, como la regresión lineal. Por ejemplo, si una variable en tu modelo es el tamaño de una casa (que puede ir de 50 a 500 metros cuadrados) y otra es el número de habitaciones (de 1 a 10), el tamaño de la casa puede tener más peso en el modelo solo porque sus valores numéricos son más grandes. La normalización hace que todas las variables se ajusten a un rango común, asegurando que cada una tenga un impacto adecuado en el modelo.

Otra ventaja de la normalización es que ayuda a que los algoritmos de optimización funcionen mejor. Estos algoritmos, como el descenso de gradiente, intentan minimizar la diferencia entre las predicciones del modelo y los valores reales. Si tienes variables con escalas muy diferentes, el algoritmo puede ajustar demasiado las variables con valores más grandes y muy poco las que tienen valores más pequeños. Por ejemplo, si estás trabajando con datos de propiedades que incluyen el tamaño del terreno (de 100 a 10,000 m²) y el número de pisos (de 1 a 5), sin normalización, el tamaño del terreno influye más en el modelo solo porque sus valores son más grandes.

Cuando el algoritmo intenta minimizar esta diferencia, puede terminar ajustando demasiado las variables con valores grandes y poco las de valores pequeños. Esto se debe a que la escala de las variables afecta cómo el algoritmo hace los ajustes.

**Usos de la normalización:**

Con respecto a los posibles usos de la normalización, un uso importante de la normalización es en modelos de machine learning que dependen de la distancia entre puntos de datos, como el algoritmo K-Nearest Neighbors. Aquí, la normalización asegura que todas las variables contribuyan por igual al cálculo de distancias, evitando que una variable con un rango amplio domine las decisiones del modelo. Por ejemplo, en un análisis de permisos de construcción, donde el salario de los trabajadores y el costo del proyecto pueden tener escalas muy diferentes, la normalización asegura que ambos se consideren de manera justa.

Además, la normalización es esencial en técnicas de reducción de dimensionalidad como el Análisis de Componentes Principales (PCA). PCA encuentra las principales direcciones en las que los datos varían y reduce la cantidad de variables necesarias, sin perder demasiada información. Al normalizar los datos, PCA puede capturar mejor la estructura de los datos, sin que las diferencias de escala entre variables afecten el análisis.

También, la normalización es útil en la visualización de datos. Al crear gráficos, como diagramas de dispersión, la normalización ayuda a comparar variables con diferentes unidades de medida de manera clara. Esto facilita ver patrones y relaciones en los datos que podrían estar ocultos debido a las diferencias de escala.

# 6. Comparación de inconsistencias

Finalmente, se utilizó la librería de ‘Data Profiling’ para generar un informe completo sobre el dataset, el cual permite comparar las inconsistencias encontradas manualmente con lo que el informe automatizado presenta. Este informe automatizado es una herramienta útil para obtener una visión rápida y detallada de los posibles problemas en los datos que pueden haber pasado desapercibidos en la revisión manual.

Además de generar un informe del dataset modificado (“ProfilingDatasetNuevo\_html.html”), se generó un informe del dataset original (“ProfilingDatasetOriginal\_html.html”) para comparar con lo que se había detectado manualmente.