Predicción Plazo de Realización en Obras Públicas de la Ciudad de Buenos Aires

Alan Barberá, Ailén Rocío Kot,
Micaela Tokashiki
Universidad Tecnológica Nacional,
Regional Buenos Aires

***Abstract*—En el presente reporte se analizarán las obras públicas que han sido finalizadas dentro del territorio de la Ciudad Autónoma de Buenos Aires (CABA) en los últimos 6 años.**

Keywords—Obras públicas, construcción, infraestructura

I. INTRODUCCIÓN Y OBJETIVOS

En el panorama actual en el que nos encontramos, en donde los datos abundan, resulta un punto estratégico poder utilizarlos para la toma de decisiones fundamentada en algo más que en la intuición, es decir, fundamentada en evidencia precisa y concreta.

Por ello es que surge el siguiente trabajo, cuya finalidad es predecir el plazo de realización de distintos tipos de obras públicas en el territorio de CABA. De esta forma, buscaremos conclusiones para facilitar la programación y planificación de nuevas obras a futuro, partiendo de algunas características de la misma, como el monto de la empresa contratista, la ubicación y el tipo de trabajo a realizar.

II. DESCRIPCIÓN DEL DATASET

A. *Observatorio de Obras Urbanas*

El dataset utilizado para el presente trabajo fue extraído del portal de datos abiertos Buenos Aires Data, en la rama de urbanismo y territorio. El mismo tiene una frecuencia de actualización mensual. Para el desarrollo del reporte se obtuvo la versión actualizada a agosto 2020.

La fuente primaria y la mantención del dataset está a cargo de la Secretaría General y Relaciones Internacionales, la Subsecretaría de Gestión Estratégica y Calidad Institucional y la Dirección General de Calidad Institucional y Gobierno Abierto.

Los datos iniciales al momento de realización del presente informe están compuestos por 1117 muestras o registros de obras públicas que pueden encontrarse finalizadas, en ejecución o en licitación. Además, cuenta con 36 features o características iniciales, entre ellas el beneficiario, el responsable, el nombre de la obra, el estudio ambiental, etc.

El portal de datos abiertos presenta una breve descripción sobre cada una de las features o características del dataset. De los cuales serán relevantes para nuestro informe los mencionados a continuación.

- Etapa (texto), indica el estado de evolución de la obra
- Tipo (texto), indica el tipo de obra
- Monto_contrato (número entero), monto en pesos por el cual se firmó el contrato
- Comuna (número entero), indica el número de comuna en donde se encuentra localizada la obra
- Plazo_meses (número entero) que es el plazo en el cual se concretará la obra

III. ANÁLISIS EXPLORATORIO DE DATOS

Luego de importar el dataset y visualizar su dimensión inicial (1117 x 36) lo primero que se realiza es analizar la existencia de valores nulos o NaNs (Not A Number) existentes. Comenzaremos analizando cada una de las features mencionadas anteriormente y entendiendo cómo está conformado el dataset para poder definir la posición a tomar frente a los datos nulos. A continuación, las conclusiones por feature:

A. *Etapa*

Las obras registradas pueden encontrarse en 3 etapas: finalizadas, en ejecución o en licitación. Esta feature no posee datos nulos.

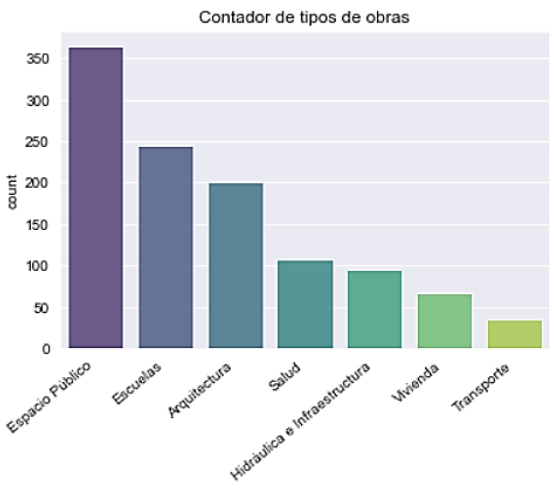
Con el presente informe se busca lograr un modelo de aprendizaje supervisado tomando como base el tiempo de realización de las obras, por lo tanto, la primer limitante que se encuentra es que se requiere que la obra haya sido finalizada para obtener los meses de duración de la obra, las cuales representan el 94% del dataset.

etapa	tipo	Porcentaje
Finalizada	1052	94.180842
En ejecución	61	5.461056
En licitación	4	0.358102

B. *Tipo*

En total hay 7 tipos de obras públicas registradas: escuelas, espacio público, vivienda, salud, arquitectura, hidráulica e infraestructura y transporte. Esta feature no posee valores nulos.

En la siguiente figura observamos cómo se distribuyen las muestras entre los distintos tipos de obras



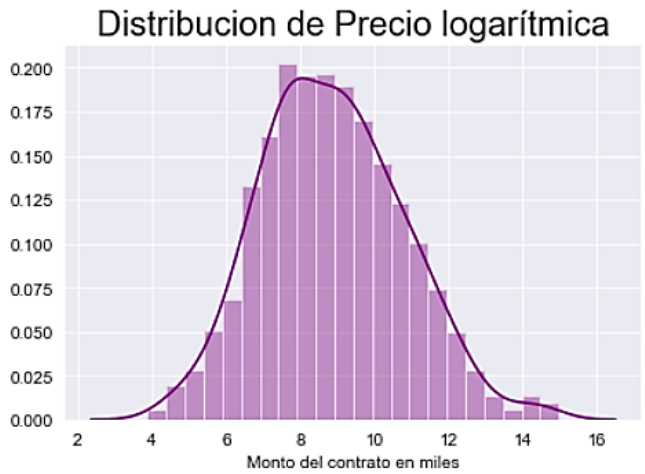
C. *Monto del contrato*

En total hay 70 samples que no poseen el detalle del monto del contrato, de las cuales, 60 de ellas corresponden a obras ya finalizadas.

Se contempló la opción de reemplazar los valores ausentes por valores representativos como la media o la mediana de la feature en función del tipo de obra, para evitar una disminución de datos. Sin embargo, se observó una gran dispersión en los montos del contrato debido a la gran amplitud entre el valor mínimo y máximo que puede obtener cada obra, por ello se decidió eliminar los valores nulos.

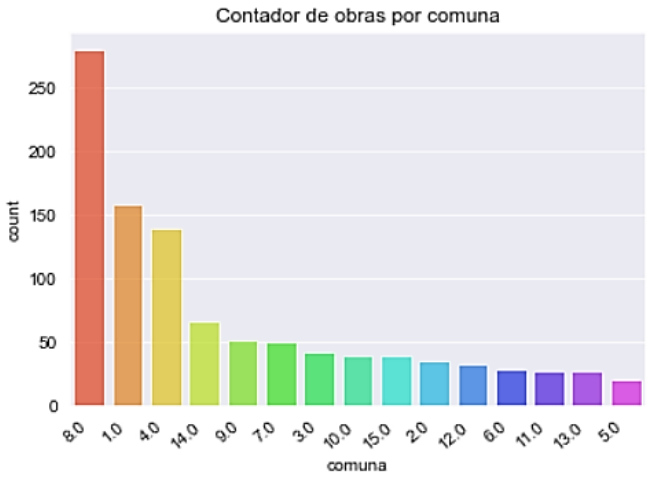
Para facilitar la visualización de este dato, también resulta conveniente generar una nueva variable en la que se guarde el monto del contrato expresado en miles.

Visualizando la distribución del monto de contrato (normalizado logarítmicamente) se observa una figura similar a la Campana de Gauss



D. *Comuna*

La comuna hace referencia a la ubicación geográfica de la obra, por ello, es un dato que no puede ser estimado o aproximado. Al igual que con el monto del contrato se define eliminar aquellas samples cuyo valor de comuna sea nulo. Una alternativa en esta instancia sería realizar un modelo de clustering para asignar un número de comuna, sin embargo, esta opción fue descartada porque la misma también está sujeta a cierto error e incluirla en este momento del análisis implicaría un error aún mayor en la predicción del objetivo del trabajo.



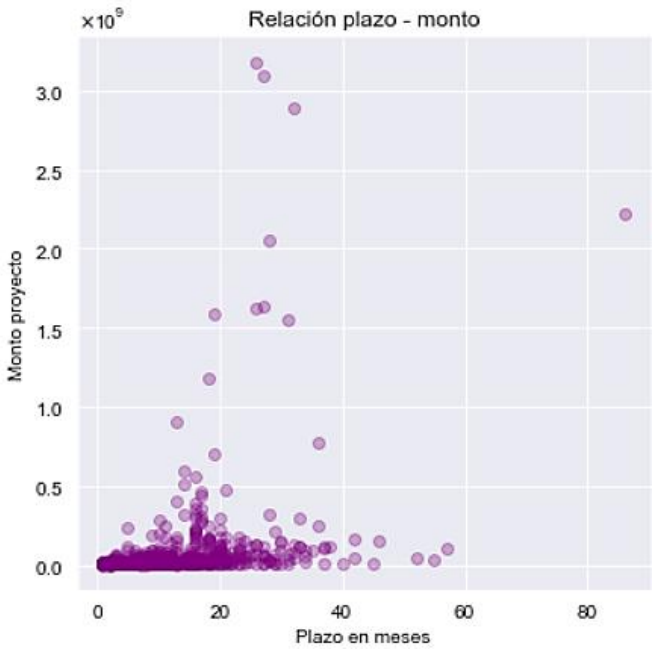
E. *Plazo en meses*

Esta feature resulta crítica ya que será el target de nuestro modelo de aprendizaje supervisado. El mismo está compuesto por 63 NaNs y 44 samples cuyo período de realización es 0 (cero) meses dentro de obras finalizadas.

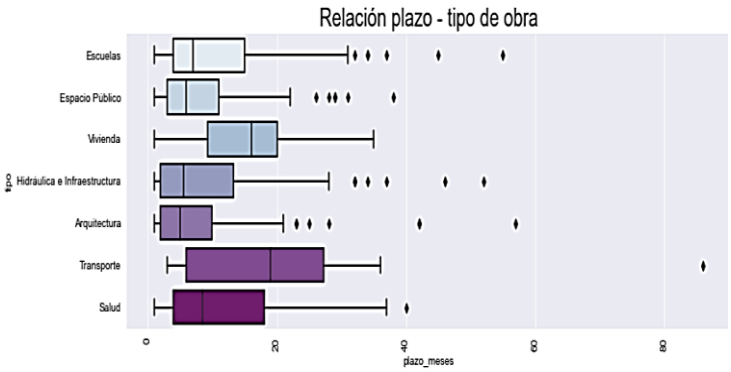
Siendo este el dato que utilizaremos más adelante para entrenar a nuestro modelo de machine learning, consideramos que aplicar un método de clustering (por ejemplo) para estimar el plazo aumentaría el error, por ello se decide eliminar los valores NaNs. En el caso de aquellas muestras con plazo 0 también se decidió eliminarlas ya que consideramos poco preciso estimar que una obra será realizada en 0 meses. Esto podría llevarnos a plantear una mejora en la recopilación o toma de datos, de incluir un campo para poder predecir con mayor precisión el tiempo en obras menores a futuro.

El dato, originalmente está expresado en números, indicando la cantidad de meses en los que fue realizada la obra. Este punto será tratado en detalle al aplicar los modelos de machine learning ya que se analizará la posibilidad de realizar la predicción por períodos, es decir, que la obra finalice en un plazo menor a 3 meses, a 6 meses, etc.

Para concluir con el EDA, observamos que en la relación entre el plazo en meses y el monto del contrato no se observa una tendencia clara.



Realizamos también un boxplot para ver cómo se relacionan el plazo con el tipo de obra y tampoco se puede extraer una conclusión en particular sobre su comportamiento



IV. MATERIALES Y MÉTODOS

A. Clasificación

Para poder predecir el plazo de realización de la obra a partir de conocer el monto del contrato, el tipo de obra y su ubicación, utilizaremos estrategias de aprendizaje supervisado (labels conocidos) a partir de modelos clasificadores.

Antes de continuar, es importante destacar que las acciones mencionadas a continuación (pre-procesamiento de datos y posterior aplicación de modelos) se realizará para el dataset con label en meses, en años y por trimestres (solamente separando al primer año en trimestres ya que es en ese plazo donde se concentraban la mayor cantidad de muestras).

El primer paso es separar del dataset nuestros targets (en una variable llamada Y) que en el caso de ser por años o por trimestres se aplicará un algoritmo llamado Label Encoder cuya función es transformar las etiquetas en valores entre 0 y n-1 clases. De la misma forma, se guardarán en una variable llamada X las features restantes que nos aportarán la información para realizar la clasificación. Nuevamente, aquellas cuyos valores sean string o texto deberán ser procesadas mediante un algoritmo de la librería Pandas, llamado get_dummies, que genera variables categóricas binarias creando nuevas features por cada clase encontrada y le asigna un valor 0 o 1, dependiendo si corresponde o no a dicha clase.

El siguiente paso es dividir el dataset en train y test, para que el modelo aprenda los mejores parámetros de la función clasificadora utilizando las muestras de entrenamiento y luego evaluarlas comparando cómo predicen muestras a las cuales nunca tuvo acceso (test set). Para esto necesitamos escalar los datos, en este caso se utilizó el método de Standard Scaler, que básicamente al valor de la muestra le resta el valor de la media y lo divide por su desvío estándar para obtener una muestra normalizada.

Para obtener los mejores hiperparámetros (restricciones definidas por el usuario) que aplicados al modelo clasificador logran una mejor precisión se utilizará el método de Cross Validation (CV) junto con Grid Search (GS). Este método consiste en dividir el set de entrenamiento en K partes (dentro de las cuales también habrá una parte de entrenamiento y otra de validación) e iterar sobre el mismo K veces con los hiperparámetros definidos en el GS.[1]

Se utilizarán los siguientes 4 modelos clasificadores[2]:

- SVC, Support Vector Classifier. Este modelo busca obtener el hiperplano separador que maximice el margen entre clases.

  Si bien se trata de un clasificador lineal, cuando las clases no son linealmente separables se puede utilizar el "kernel trick" que son funciones de similaridad entre muestras. Básicamente, el kernel traslada los datos a un nuevo espacio de alta dimensión en donde sí son linealmente separables.

  En el trabajo práctico, el kernel que mejor se adaptó fue el RBF (Radial Basis Function) o kernel gaussiano. El cual se define como:

$$K_{Gaussiano}\,(Xi\,,Xj) = exp\ exp\ (-\frac{\|Xi - Xj\|^2}{2\sigma^2})$$

- KNN, K-Nearest Neighbour Classifier. Este modelo asigna la etiqueta a cada sample en función de la proximidad que presenten entre sí, teniendo en cuenta las etiquetas de los "K-vecinos" más próximos, calculando la distancia del elemento nuevo a cada uno de los existentes.

$$d(P1, P2) = \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2}$$

- Random Forest Classifier. Este modelo se basa en la construcción de árboles de decisión. Una característica que los distingue es que son modelos muy buenos para capturar interacciones complejas. Es por esto que en la mayoría de los casos se posicionó dentro de los mejores clasificadores.

- Logistic Regression. Este modelo también es un clasificador lineal, su estructura es la de una regresión lineal precedida por una función de activación sigmoide lo cual le permite generar un output binario y no continuo.

  La función sigmoide es la siguiente:

$$f(x) = \frac{1}{1 + e^{-x}}$$

  Si reemplazamos al término x por una ecuación lineal, se obtiene la regresión logística.

Para evaluar la performance de los modelos nos basaremos en el accuracy o precisión.

El **accuracy** se obtiene sumando todas las clases correctamente clasificadas y dividiendo por el total de muestras (train y test)

Y una **matriz de confusión** para visualizar la clasificación por clase.

Para el caso del análisis efectuado en los datasets por años y por trimestres vamos a observar que se trata de un set desbalanceado. Esto significa que de ciertas categorías posee pocas muestras, lo cual reduce el accuracy ya que el modelo tiene pocas muestras de ese estilo para poder entrenar y definir los mejores hiperparámetros. Una forma de solucionar este problema es mediante el método de Resample [3][4], que básicamente asigna más muestras similares a aquellas clases con menor cantidad para balancear el set. Esto presenta un mayor costo computacional al aumentar la dimensión de los datos a entrenar y testear.

V. RESULTADOS

Con las herramientas antes descriptas obtuvimos para el análisis en meses un accuracy máximo del 15% con el método de Random Forest Classifier.

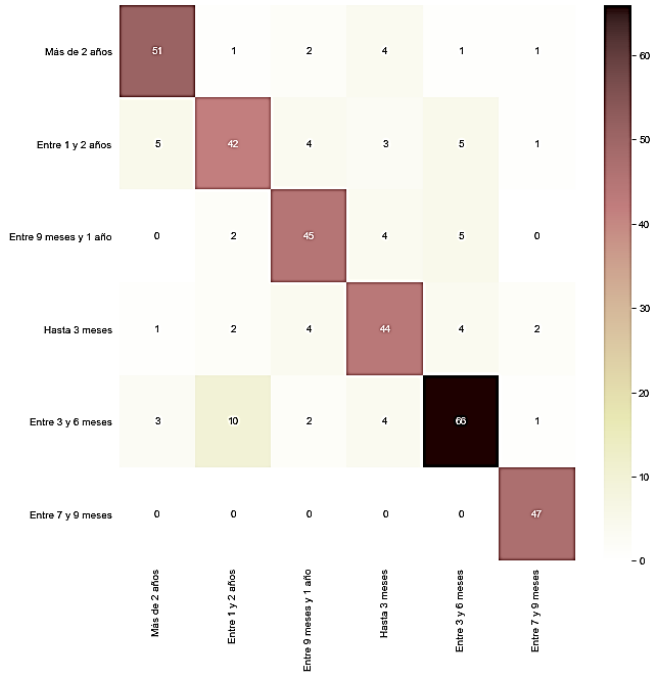
	Clasificador	Accuracy
0	SVC en meses	0.090468
1	KNN en meses	0.134087
2	Random Forest en meses	0.150242
3	LR en meses	0.108239

Con el set de plazos clasificados en años, el máximo accuracy obtenido (en muestras con resample) fue de 84% también con el método de Random Forest.

	Clasificador	Accuracy
0	SVC en años	0.712173
1	KNN en años	0.800910
2	Random Forest en años	0.847554
3	LR en años	0.635950

Y, finalmente, clasificando por trimestres en el primer año y por años en los restantes, el mejor accuracy (en muestras con resample) fue del 80% bajo el método de KNN, ligeramente superior que el Random Forest Classifier.

	Clasificador	Accuracy
0	SVC en trim, sin resample	0.321075
1	KNN en trim, sin resample	0.340877
2	Random Forest en trim, sin resample	0.356436
3	LR en trim, sin resample	0.384724
4	SVC en trim, con resample	0.609290
5	KNN en trim, con resample	0.806011
6	Random Forest en trim, con resample	0.797814



VI. DISCUSIÓN Y CONCLUSIONES

Dentro de las conclusiones obtenidas durante el desarrollo del presente trabajo destacamos que, aunque poseamos una cantidad de muestras considerable (más de 800 samples), al clasificar entre muchas categorías (más de 30), como en el caso de la predicción en meses, la calidad del modelo puede ser baja por caer en la problemática de que el dataset esté desbalanceado.

Con los modelos restantes los resultados obtenidos fueron notablemente superiores por la utilización de la herramienta resample, sin embargo, esto tiene un costo computacional grande, llevando a que el tiempo de entrenamiento de cada modelo puede llegar a demorar horas.

Finalmente, desde el punto de vista del grupo, consideramos que la mejor alternativa de las 3 analizadas es la clasificación en períodos trimestrales. Si bien el accuracy en períodos anuales fue superior, se pierde nivel de detalle. Consideramos más útil poder estimar y obtener información para la toma de decisiones si el rango de realización es de 3 meses.

Como consideración o recomendación para trabajar a futuro, sería en especificar aún más en detalles de la obra al momento de recolectar información, como por ejemplo, si el período es menor a 1 mes, indicar las semanas utilizadas.

REFERENCES

[1] VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.".

[2] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

[3] Javaid Nabi. "Machine Learning – Multiclass Classification with Imbalanced Dataset." (2018)
URL:https://towardsdatascience.com/machine-learning-multiclass-classification-with-imbalanced-data-set-29f6a177c1a

[4] Elite Data Science. "How to Handle Imbalanced Classes in Machine Learning"
URL: https://elitedatascience.com/imbalanced-classes