TECNOLÓGICO DE MONTERREY®

Inteligencia artificial avanzada para la ciencia de datos

Reto

Etapa 4. Reporte de modelación

Diego Arturo Padilla Domínguez - A01552594

Keyuan Zhao - A01366831

Carolina Herrera Martínez - A01411547

Cutberto Arizabalo Nava - A01411431

Jose Pablo Cobos Austria - A01274631

Campus Querétaro

04 de noviembre de 2022

Etapa 4: “Modeling”

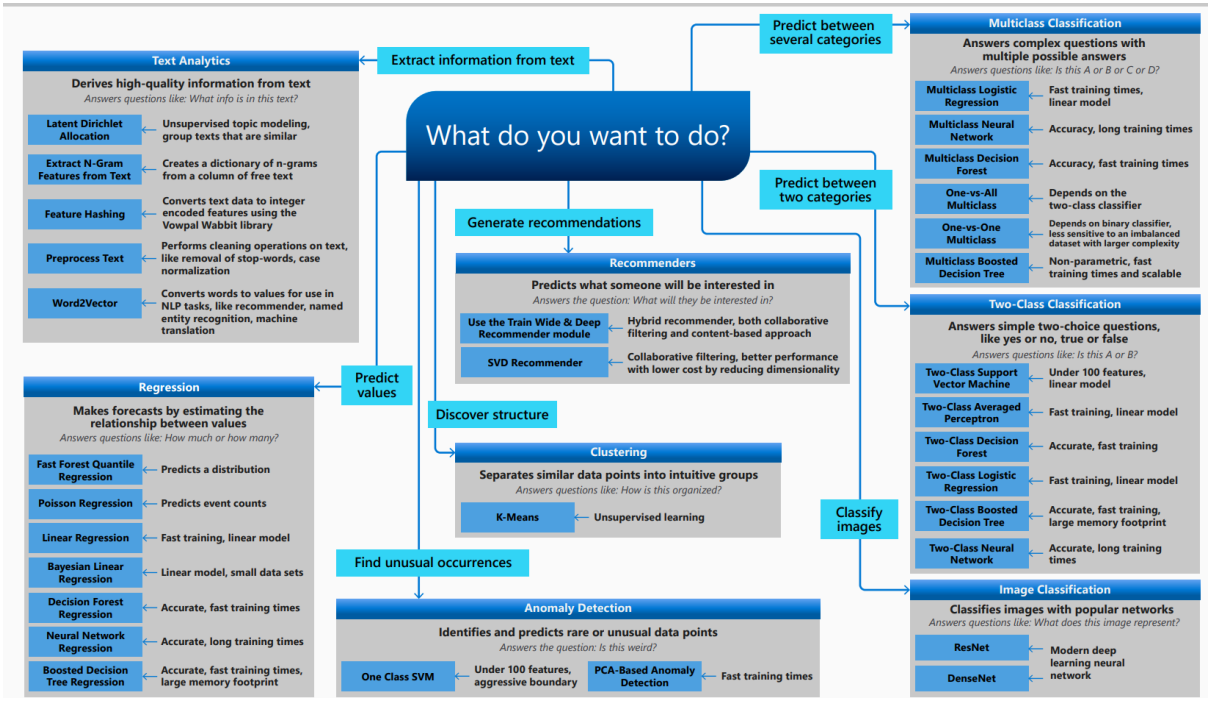
Para ejecutar esta etapa es esencial enfocarnos en el objetivo de la minería de datos, el cual es el siguiente:

“Obtener datos representativos sobre las rutas Origen-Destino mediante la limpieza, transformación y análisis del set de datos inicial. Utilizar dichos datos para entrenar un modelo que sea capaz de predecir la cantidad de viajes a una comuna en base a las características de dicha comuna.“

Por ende, resulta natural que una vez obtenida la matriz de viajes procedamos a buscar cuál es el mejor modelo para dicha predicción.

En el área de ML existen 3 grandes ramas con un gran abanico de modelos que podemos utilizar. Por lo tanto, es importante saber distinguir qué es lo que queremos hacer, para poder elegir el modelo adecuado.

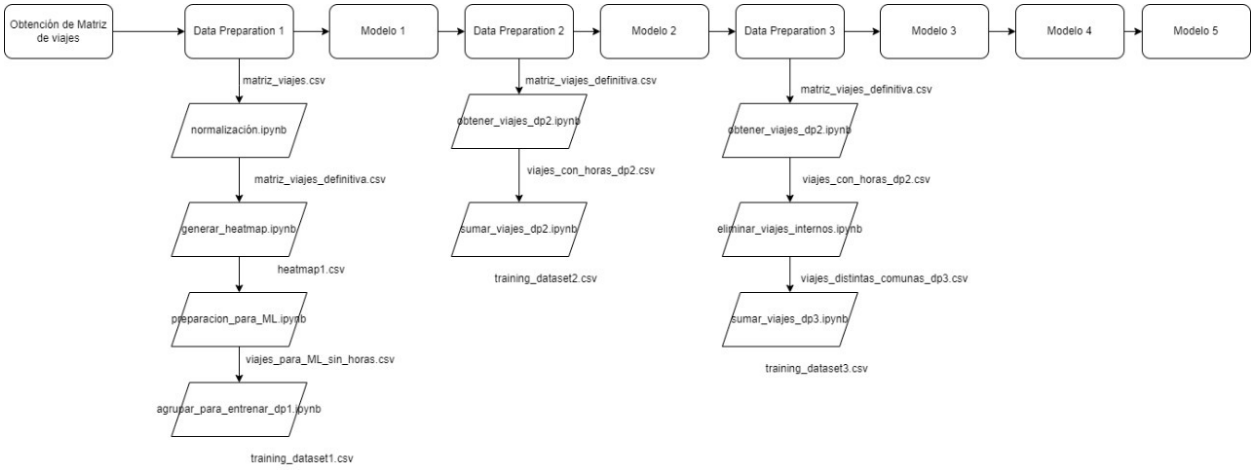
Como base para esta decisión, usaremos la guía de ML de Microsoft Azure, la cual se muestra a continuación:



En nuestro caso, queremos predecir valores (cantidad de visitas a una comuna) en base a los atributos de la comuna, por lo que necesitamos usar una Regresión.

Vemos que existen distintos tipos de regresión, cada tipo tiene distintas ventajas y requisitos para su uso.

Diagrama de modelación



Diseño de pruebas

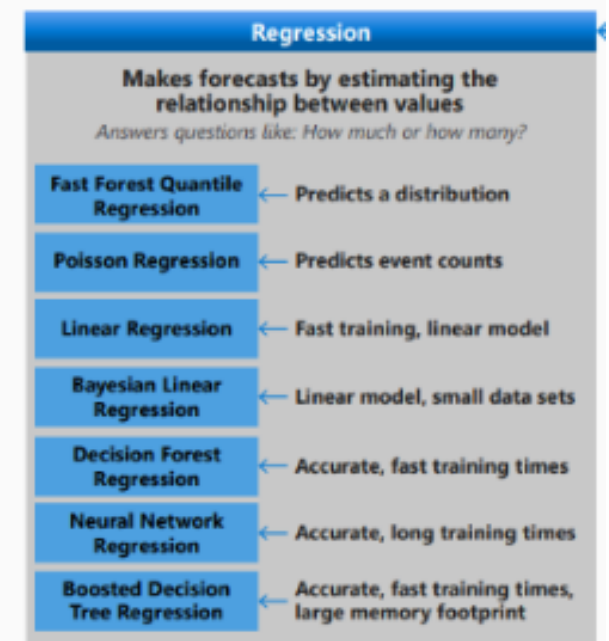
Debido a la cantidad de registros se decidió hacer una distribución del dataset en:

- Entrenamiento: 85%
- Pruebas: 15%

Para verificar la calidad y validez de los datos se tomará como medida el porcentaje de variación entre los valores reales y calculados del dataset de pruebas.

Para cada uno de los modelos se usarán las mismas pruebas.

Modelo 1. Regresión Lineal



Como el modelo más sencillo de regresión es el de regresión lineal, lo utilizaremos como modelo base/benchmark, y este será nuestro punto de comparación con el resto de modelos.

Además, el modelo de regresión lineal nos brinda la ventaja de que podemos analizar las propiedades estadísticas para obtener insights como la significancia de cada variable independiente.

Framework: Sci-Kit Learn

Descripción de datos:
Este modelo recibe como variables independientes lo siguiente:

- Número de escuelas
- Número de hospitales
- Número de iglesias
- Número de zonas típicas

Y como variable dependiente:

- Número de viajes hacia la comuna

Parámetros:
Al ser un modelo de regresión lineal y se utilizará como modelo benchmark, este utiliza los parámetros por defecto de la librería.

Descripción del modelo:
Se esperan resultados no muy significativos, se estima una variación promedio mayor al 50%.

No se puede modelar relaciones complejas y no se pueden capturar relaciones no lineales sin transformar la entrada, por lo que se tiene que trabajar duro para que se ajuste a funciones no lineales. Puede sufrir con valores atípicos.

Evaluación del modelo:
El modelo obtuvo una variación del 67.03%, este resultado está altamente alejado de nuestro objetivo de minería de datos, pero dando los resultados esperados previos al entrenamiento del modelo.

```
dataframe = pd.DataFrame()
dataframe['Num_viajes'] = y_test
dataframe['Num_viajes_p'] = ypred_linear
dataframe['dif'] = abs(dataframe['Num_viajes'] - dataframe['Num_viajes_p'])
import math
dataframe['Num_viajes_p'] = dataframe['Num_viajes_p'].apply(lambda x: math.ceil(x))
dataframe["Dif_perc"] = dataframe["dif"] / dataframe["Num_viajes"] *100
print("Variación de valor real contra predecido: ", dataframe["Dif_perc"].mean(), "%")
```

```
Variación de valor real contra predecido:  67.03080147997053 %
```

Modelo 2. Regresión Lineal

Debido a que se realizó una añadidura en la cantidad de variables, se utilizará nuevamente el modelo inicial de regresión lineal con el fin de saber si las variables añadidas son útiles para la mejora del modelo.

Framework: Sci-kit Learn
Este modelo recibe como variables independientes lo siguiente:

- Número de escuelas

- Número de hospitales
- Número de iglesias
- Número de zonas típicas
- 23 variables dummies para las horas del día

Y como variable dependiente:

- Número de viajes hacia la comuna

Parámetros:
Al ser un modelo de regresión lineal y solo se busca ver si las variables agregadas mejoran el modelo, éste utiliza los parámetros por defecto de la librería.

Descripción del modelo:
Se espera una mejora de al menos un 10% con respecto al modelo anterior.

Al igual que en el modelo anterior, no se pueden modelar relaciones complejas. No se pueden capturar relaciones no lineales sin transformar la entrada, por lo que se tiene que trabajar duro para que se ajuste a funciones no lineales. Puede sufrir con valores atípicos.

Evaluación del modelo:
El modelo obtuvo una variación del 54.92%, por lo cual no cumple con nuestro objetivo de minería de datos. A comparación del modelo anterior se obtuvo una mejora del 12%.

```
dataframe = pd.DataFrame()
dataframe['Num_viajes'] = y_test
dataframe['Num_viajes_p'] = ypred_linear
dataframe['dif'] = abs(dataframe['Num_viajes'] - dataframe['Num_viajes_p'])
import math
dataframe['Num_viajes_p'] = dataframe['Num_viajes_p'].apply(lambda x: math.ceil(x))
dataframe["Dif_perc"] = dataframe["dif"] / dataframe["Num_viajes"] *100
print("Variación de valor real contra predecido: ", dataframe["Dif_perc"].mean(), "%")
```

```
Variación de valor real contra predecido:  54.92606358208957 %
```

Modelo 3. Regresion lineal

Debido al cambio de objetivo de negocio y de minería de datos se tiene que realizar un nuevo modelo benchmark, escogiendo nuevamente el modelo de regresión lineal por su simpleza para la interpretación.

Framework: Sci-kit Learn

Este modelo recibe como variables independientes lo siguiente:

- Número de escuelas
- Número de hospitales

- Número de iglesias
- Número de zonas típicas
- 23 variables dummies para las horas del día

Y como variable dependiente:

- Número de viajes hacia la comuna

Parámetros:
Como modelo benchmark se utilizan los parámetros por defecto de la librería.

Descripción del modelo:
Para este modelo se tiene esperado una variabilidad mayor a un 50%.

Recordando nuevamente las limitaciones como lo es no poder modelar relaciones complejas. No se pueden capturar relaciones no lineales sin transformar la entrada, por lo que se tiene que trabajar duro para que se ajuste a funciones no lineales. Puede sufrir con valores atípicos.

Evaluación del modelo:
El modelo obtuvo una variación del 89.3 %, este resultado está altamente alejado de nuestro objetivo de minería de datos. Este modelo brinda los resultados esperados al momento de su elección como modelo benchmark.

```
dataframe = pd.DataFrame()

dataframe['Num_viajes'] = y_test

dataframe['Num_viajes_p'] = ypred_linear

dataframe['dif'] = abs(dataframe['Num_viajes'] - dataframe['Num_viajes_p'])

import math

dataframe['Num_viajes_p'] = dataframe['Num_viajes_p'].apply(lambda x: math.ceil(x))

dataframe["Dif_perc"] = dataframe["dif"] / dataframe["Num_viajes"] *100

dataframe["Dif_perc"].mean()
```

```
89.33346484561275
```

Modelo 4. Random forest

Los modelos de decisión basados en bosques (conjuntos de árboles) se caracterizan por ser más precisos que los modelos sencillos generados a base de regresiones lineales. Además, estos modelos nos permiten conocer la significancia de cada variable que integra

al modelo, por lo que nos resultará de gran utilidad para cumplir con el objetivo de negocio de conocer cuales son los principales atractores de viajes.

Framework: Sci-kit Learn

Este modelo recibe como variables independientes lo siguiente:

- Número de escuelas
- Número de hospitales
- Número de iglesias
- Número de zonas típicas
- 23 variables dummies para las horas del día

Y como variable dependiente:

- Número de viajes hacia la comuna

Parámetros:
Debido a que este modelo se utilizará únicamente como un paso intermedio en la mejora del rendimiento, se utilizarán los valores por default del regresor de random forest de sklearn, es decir:

- n_estimators: 100
- max_depth : None
- min_samples: None

Descripción del modelo:
Para este modelo se tiene esperado una variabilidad alrededor de un 30%.

Las desventajas de este modelo es que se pierde la facilidad de interpretación del modelo, además de que no puede predecir más allá del rango de valores del conjunto de entrenamiento.

Evaluación del modelo:
El modelo obtuvo una variación del 15.05 %, el resultado es casi aceptable para el objetivo de mineria,

Los resultados obtenidos son muy superiores a los esperados, el modelo superó al anterior en un 74% siendo una mejora bastante significativa.

```
import  math

dataframe['Num_viajes_p']  =    dataframe['Num_viajes_p'].apply(lambda  x:  math.ceil(x))

dataframe["Dif_perc"]  =    dataframe["dif"]  /  dataframe["Num_viajes"]  *100

dataframe["Dif_perc"].mean()

15.046401407989643
```

Modelo 5. XGBoost

El modelo XGBoost nos ofrece un rendimiento superior a los modelos de árboles tradicionales gracias a que implementa una optimización avanzada utilizando Gradient Boosting.

Framework: Sci-kit Learn

Este modelo recibe como variables independientes lo siguiente:

- Número de escuelas
- Número de hospitales
- Número de iglesias
- Número de zonas típicas
- 23 variables dummies para las horas del día

Y como variable dependiente:

- Número de viajes hacia la comuna

Parámetros:
Para afinar los hiperparámetros de este modelo, recurrimos a utilizar una búsqueda en cuadrícula CV. Este enfoque nos permite probar diferentes valores de los hiperparámetros para encontrar los que ofrecen el mejor rendimiento.

Dándonos como resultado los siguientes hiperparametros:

- colsample_bytree: 0.1
- learning_rate: 0.1
- max_depth: 6
- n_estimators: 100000

Descripción del modelo:
Para este modelo se tiene esperado una variabilidad menor al 15%

Las limitaciones de este modelo es que sus resultados pueden llegar a ser complejos de interpretar, puede llegar a ajustar ciertos grupos de datos en presencia de ruido y se tiene poco control sobre lo que hace el modelo.

Evaluación del modelo:

El modelo obtuvo una variación del 13.06 %, el resultado cumple con creces el objetivo de la minería de datos, dando un resultado un 2% por encima de lo esperado.