# Proyecto final

## 1 Información del Dataset

El dataset MajorCityDF tiene información sobre fechas (dt), temperatura promedio (AverageTemperature), incertidumbre de la temperatura promedio (AverageTemperaturUncertainy), pais (Country), ciudad (City), latitud (Latitude) y longitud (Longitude). La fecha viene en formato date32, los datos de temperatura son float y los demás datos son strings. Se crearon dos nuevas variables en formato float, año y mes, a partir de la fecha. Tomando en consideración estas nuevas columnas, MajorCity esta compuesto de 9 columnas y 239177 filas. Estos datos son provistos de **challange de kaggle** Climate Change: Earth Surface Temperature Data y son construidos a partir de los datos de Berkeley Earth data page

### 1.1 Ingeniera de datos

Los datos de temperatura e incertidumbre contenían **NaN** al rededor de 11 mil datos. Se tomo la decisión de hacer una **interpolación de datos lineal** para evitar perdida de información tomando el dato siguiente y el anterior para rellenar cada NaN, respetándose el cálculo según su grupo ( ciudad ). Notamos datos desbalanceados, donde algunos países, como china e india, tenían muchos registros de temperaturas en comparación a los demás países. En otro plot de temperatura registrada por año, notamos que a medida que pasa el tiempo la temperatura aumenta, lo que nos indica que hay valores extremos, sobretodo de baja temperatura entre los años 1750 y 1850.

Para predecir la temperatura se vió como un problema de de regresión. Se utilizaron las variables latitude, longitude, year y month como fetaures, y se utilizo la temperatura promedio mensual por ciudad como label. Como para regresion se requieren variables continuas, se transformaron las variables latitude y longitude en valores float, haciendo uso de la funcion coord_change.

Para el análisis de outliers se realizó un cruce de datos para identificar las ciudades con su respectivo continente. Los datos de Latatitude y Longitude se pasaron a formato numerico para poder realizar mapas. Se codificaron las ciudades, paises y continentes con una etiqueta numérica. Se calculó la temperatura promedio de cada mes usando una media movil de 12 meses, y el promedio de cada año usando aquellos valores.

Para revisar el código revisar el repositorio en github

## 2 Modelos de ML

El objetivo de los modelos mostrados a continuación es predecir la temperatura promedio mensual en una ciudad. Se define que el 20% del dataset sera set de prueba y el resto

sera set de entrenamiento. Para ambos modelos se entregan los valores de las métricas de evoluciona MAE, MSE, RMSE y $R^2$.

## 2.1 Decission Tree (DT)

Este modelo supervisado consiste en dividir el dataset original según los valores de las variables, buscando siempre separar en dos clases (en este caso porque es un árbol binario). Se eligió este modelo, porque es fácil de visualizar la toma de decisiones. En la Figura 1, el gráfico arriba a la izquierda muestra la curva de aprendizaje de para un DT sin poda y el gráfico de arriba a la derecha muestra la curva de aprendizaje para un DT con poda de tipo mínimo de muestras por hoja. Al agregar poda al DT, se observa que la varianza entre la curva de entrenamiento y prueba disminuye, lo que implica un modelo menos complejo y una mejora desde el DT original.

## 2.2 KNN

Este modelo consiste en definir el valor de un nuevo punto, en este caso su temperatura promedio, basándose en el valor de los vecinos cercanos. Este modelo es sensible a los datos no estandarizados, sin embargo lo consideramos optimo para la predicción, ya que en un comienzo los valores de temperatura promedio parecían no variar mucho. En la Figura 1, se observa el el gráfico abajo a la izquierda la curva de aprendizaje de este modelo con 5 vecinos cercanos y el gráfico a la derecha es del mismo modelo pero con 10 vecinos cercanos. Notamos que entre mas vecinos la varianza disminuye y el modelo sera menos complejo.

## 2.3 Bias, varianza y funcion de costo

Para estudiar mas la performance del modelo, se calculo el bias, la varianza y la funcion de costo para ambos modelos (DT con poda y KNN con 10 vecinos). Como bias y varianza dependen del promedio de las predicciones (esperanza), se uso el modelo para 10 submuestras del set de entrenamiento generadas al azar, de cada uno se obtuvo la predicción usando el set de prueba y finalmente se calculo la funcion de costo para ambos modelos. Los valores estan presentes en el codigo.

## 2.4 Descomposición de tendencias

Se realizó una descomposición de la tendencia de las temperaturas en ruido + estacionalidad + tendencia. La tendencia daba positiva al alza de la temperatura a lo largo de los años, la estacionalidad estaba macada por la estacionalidad del año (otoño, invierno, primavera y verano), y el ruido era completamente aleatorio, donde el ruido inicial es mayor dado a la mayor incertidumbre de los datos, sin embargo el ruido tenía estructuras locales, las cuales son interesantes de analizar, ya sea con Isolation Forest o z-score. Donde se hipotetiza que se debe a eventos puntuales y naturales.

### 2.5 Isolation Forest

Isolation Forest es un algoritmo de detección de anomalías basado en el principio de que los puntos atípicos son más fáciles de "aislar" que los puntos normales. El método construye múltiples árboles binarios aleatorios (llamados iTrees) donde, en cada división, selecciona una característica y un punto de corte de manera aleatoria.

Los datos que requieren pocas divisiones para quedar aislados se consideran anómalos, mientras que aquellos que necesitan muchas divisiones se clasifican como normales. La anomalía se determina calculando la longitud promedio del camino necesario para aislar cada observación en todos los árboles. Este enfoque es eficiente, escalable para grandes volúmenes de datos y no requiere suponer una distribución específica de los datos.

Se utilizó este modelo para detectar outliers en la serie temporal de temperaturas MA12 (media movil) por ciudad. Se comparó con el método clásico "z-score". Este fue capáz de detectar outliers de diferentes partes de la serie temporal sinq ue otros outliers los oculataran, la cual es la deblidad de z-score. Tmabién se utilizó para analizar el ruido de la descomposición de tendencias y se comparó con la actividad volcánica de oceanía.

### 2.6 DBscan

HDBSCAN es un algoritmo de agrupamiento basado en densidad que extiende a DBSCAN incorporando una estructura jerárquica. Construye un árbol de clusters a partir de las variaciones de densidad en los datos y luego selecciona automáticamente la partición más estable.

A diferencia de métodos tradicionales como k-means, HDBSCAN no requiere especificar el número de clusters y es capaz de identificar grupos de distintas formas y densidades, además de clasificar puntos que no encajan en ningún grupo como ruido.

Se utilizó para detectar outliers basaádose en las densidades locales de la serie, la cual se tuvo que normalizar para que pudiera hacer bien el análisis, donde la distancia puede afectar al resultado.

## 3 Conclusión

Nos parece que los modelos elegidos fueron apropiados, pero quizás usando random forest o redes neuronales podríamos buscar una mejor performance, con menos simplificaciones. En el gif de temperaturas notamos que la temperatura global aumenta con los años. Las causas pueden ser estudiadas para un próximo proyecto, ya que incluir datos de actividad volcánica e incremento de niveles de CO2 implicaba un trabajo mas extenso.

Entre los métodos de detección de outliers Isolation Tree detectó más anomalías qye DBscan, y este más que z-score. Tiene la utilidad de poder detectar eventos extraños y anomalías climáticas que pueden estar relacionadas con eventos volcánicos, corrientes marítimas, y la producción de C02 para trabajos a futuros. Notamos que la producción de

SO2 en grandes cantidades puede bajar la temperatura para volcanes de clasificación VEI 7 de manera temporal, luego vuelve a subir la temeperatura.

Se concluye que la actividad volcánica de VEI 7 del volcán Tambora en el año 1813 fue capáz de afectar la temperatura disminuyéndola, sin embargo volcanes de VEI6 o menores no hay efecto lo suficientemente medible u observable.

# 4 Anexo

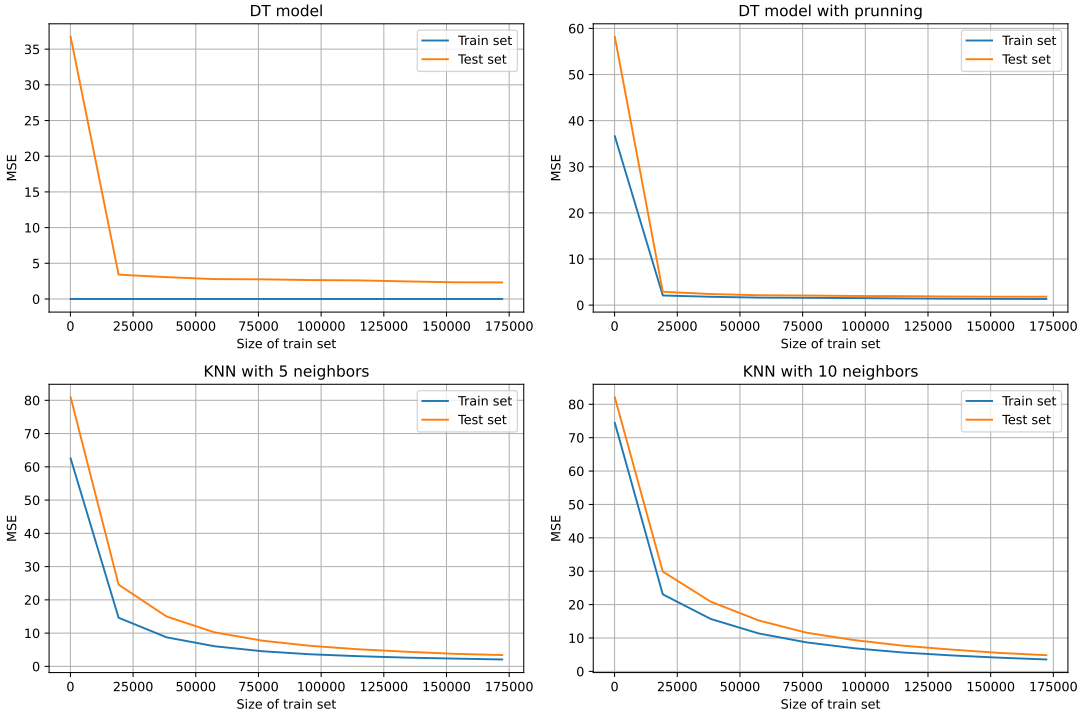


Figure 1: Curva de aprendizaje para los modelos DT y KNN con distintos parámetros de regularización.