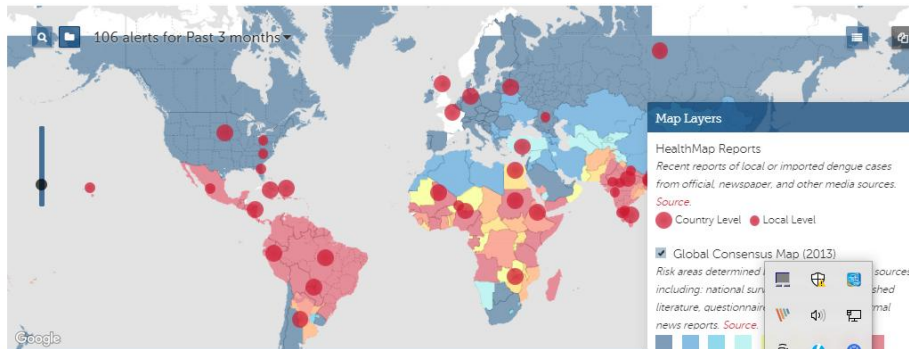


Actividad 3.6 - DengAI: predicción de la propagación de enfermedades



El objeto de esta actividad es participar en la competición de ofrecida de la web de DrivenData denominada: DengAI: Predicting Disease Spread.

Para ello accederemos a la siguiente web y nos crearemos un usuario:

Título: DengAI: Predicting Disease Spread

Url: <https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/>

DRIVEN DATA Competitions How it works Partner with us DrivenData+RBI Blog Profile Log out

HEALTH
DengAI: Predicting Disease Spread
Using environmental data collected by U.S. Federal Government agencies, can you predict the number of dengue fever cases reported each week in San Juan, Puerto Rico and Iquitos, Peru?

Intermediate practice 7 months left 14,268 joined

Navigation
Home
About
Problem description
Official rules
Leaderboard
Discussion (52)
Data download
Submissions (9)
Share your work

Problem description
Your goal is to predict the `total_cases` label for each `(city, year, weekofyear)` in the test set. There are two cities, San Juan and Iquitos, with test data for each city spanning 5 and 3 years respectively. You will make one submission that contains predictions for both cities. The data for each city have been concatenated along with a `city` column indicating the source: `sj` for San Juan and `iq` for Iquitos. The test set is a pure future hold-out, meaning the test data are sequential and non-overlapping with any of the training data. Throughout, missing values have been filled as `NaN`.

Features
[List of features](#)

On this page
The features in this dataset:
City and date indicators
NOAA's GHCN daily climate data weather station measurements
PERSIANN satellite precipitation measurements (0.25x0.25 degree scale)
NOAA's NCEP Climate Forecast System Reanalysis measurements (0.5x0.5 degree scale)
Satellite vegetation - Normalized difference vegetation index (NDVI)
- NOAA's CDR Normalized Difference Vegetation Index (0.5x0.5 degree scale) measurements

La actividad consiste en subir a dicha web un fichero csv con la estimación que hayamos obtenido al aplicar los modelos que consideres oportunos. Podrás observar que se pueden realizar hasta un máximo de 3 subidas diarias y la propia web realizará una valoración de tu solución. (La valoración se realiza utilizando el MAE como criterio de valoración de calidad)

Hay que realizar todo el proceso completo de importación del dataset, ajuste de características, selección de características, entrenamiento y selección de un modelo para entrenarlo, predicción y subida del fichero para que la web lo valore y clasifique en la competición.

Consideraciones a tener en cuenta:

- Es necesario realizar pruebas con al menos tres modelos diferentes considerando que han de utilizarse como mínimo **NaiveBayes**, **KNN** y **otro modelo** que consideres oportuno
- Realizar pruebas de hiperparametrización con las dos técnicas explicadas: GridSearch y Random Search.

Una vez hayas realizado las diferentes pruebas, has de generar un documento pdf donde:

Explices con detalle tu solución. Ha de contener capturas de tu posicionamiento en la competición con la clasificación que has obtenido en la competición con los diferentes SUBMITs realizados. Sería muy interesante que expliques las mejoras o no, que has obtenido con las diferentes pruebas.

PYTHON VS SQL

Operation	Python	SQL
Create/Load Data	<code>df = pd.read_csv('file.csv')</code>	<code>CREATE TABLE table_name (column1 datatype, column2 datatype, ...);</code>
View Data	<code>df.head()</code>	<code>SELECT * FROM table_name LIMIT 5;</code>
Get Dimensions	<code>df.shape</code>	<code>SELECT COUNT(*) FROM table_name;</code>
Show Info	<code>df.info()</code>	<code>DESCRIBE table_name; or SHOW COLUMNS FROM table_name;</code>
Summary Statistics	<code>df.describe()</code>	<code>SELECT AVG(column), MIN(column), MAX(column), COUNT(*) FROM table_name;</code>
Select Columns	<code>df[['col1', 'col2']]</code>	<code>SELECT col1, col2 FROM table_name;</code>
Filter Rows	<code>df[df['column'] > value] or df.query('column > value')</code>	<code>SELECT * FROM table_name WHERE column > value;</code>
Select Rows by Index	<code>df.iloc[row_index]</code>	<code>SELECT * FROM table_name WHERE id = row_index;</code>
Sort Data	<code>df.sort_values('column')</code>	<code>SELECT * FROM table_name ORDER BY column ASC/DESC;</code>
Group and Aggregate	<code>df.groupby('column').agg()</code>	<code>SELECT column, COUNT(*) FROM table_name GROUP BY column;</code>
Join Tables	<code>pd.merge(df1, df2, on='key')</code>	<code>SELECT a.col1, b.col2 FROM table1 a JOIN table2 b ON a.key = b.key;</code>

Fuente:

https://www.linkedin.com/feed/update/urn:li:activity:7290306350243274752/?utm_source=share&utm_medium=member_desktop

Título: Modelos de machine learning: Guía básica para principiantes

Url: <https://planetachatbot.com/modelos-de-machine-learning-guia-basica-para-principiantes/>

Criterios para valorar Proyecto/Reto	
Peso %	Tareas (Se evalúa desde 0 hasta 10)
5	Utilizar el <u>drive</u> o <u>GitHub</u> como origen de ficheros para la importación del <u>dataset</u> .
5	Importación del <u>dataset</u> : Preparación de los datos: Normaliza, ajusta la calidad de los datos.
5	Selección de <u>características</u> : Utiliza métodos no gráficos para la selección de características.
5	Selección de <u>características</u> : Utiliza herramientas gráficas para la elección de las características.
10	Además de la división de los datos de <u>train</u> y <u>test</u> , incorpora la utilización de datos de validación.
10	Entrenamiento: Modelo 1º <u>NaiveBayes</u> - Desarrolla las diversas pruebas propuestas para la selección y <u>justifica</u> el criterio de calidad para la selección del modelo. Utiliza <u>Cross Validation</u> .
10	Entrenamiento: Modelo 2º <u>KNN</u> - Desarrolla las diversas pruebas propuestas para la selección y <u>justifica</u> el criterio de calidad para la selección del modelo. Utiliza <u>Cross Validation</u> .
10	Entrenamiento: Modelo 3º de elección libre - Desarrolla las diversas pruebas propuestas para la selección y <u>justifica</u> el criterio de calidad para la selección del modelo. Utiliza <u>Cross Validation</u> .
10	Entrenamiento - Uso de gráficos: Integra el uso de gráficos para obtener comparativas en el entrenamiento de los modelos.
10	Predicción: Utiliza herramientas gráficas para ayudar a entender la precisión de los resultados obtenidos.
5	Predicción: Describe con claridad una valoración de los resultados obtenidos.
5	<u>Submit</u> del fichero con la predicción y captura de la valoración /posicionamiento obtenido en la competición.
5	Propone soluciones creativas e innovadoras.
5	El <u>pdf</u> final tiene una portada., utiliza un índice, apartado de conclusiones y referencias (web). Se hace <u>mención</u> a referencias externas, no recogidas en el material suministrado.
Total	100

Formato de entrega

Es **obligado** entregar un fichero en un Archivo PDF con capturas del código y resultados obtenidos, así como la url de GitHub y Google Colab donde has publicado el código.

- Nombrar el archivo siguiendo el siguiente patrón:

SNS_ACT3_6_NombreApellidos.pdf