

# DSI - Predicción

Luis Cabañero Gómez

# Introducción

## Competición DengAI de DrivenData

**Intentos: 24**

**Mejor puntuación: 25.2188**

# Herramientas

- **Python**
- **NumPy**
- **Pandas**
- **Scikit-Learn**
- **Keras**
  - TensorFlow

# Acercamiento al problema

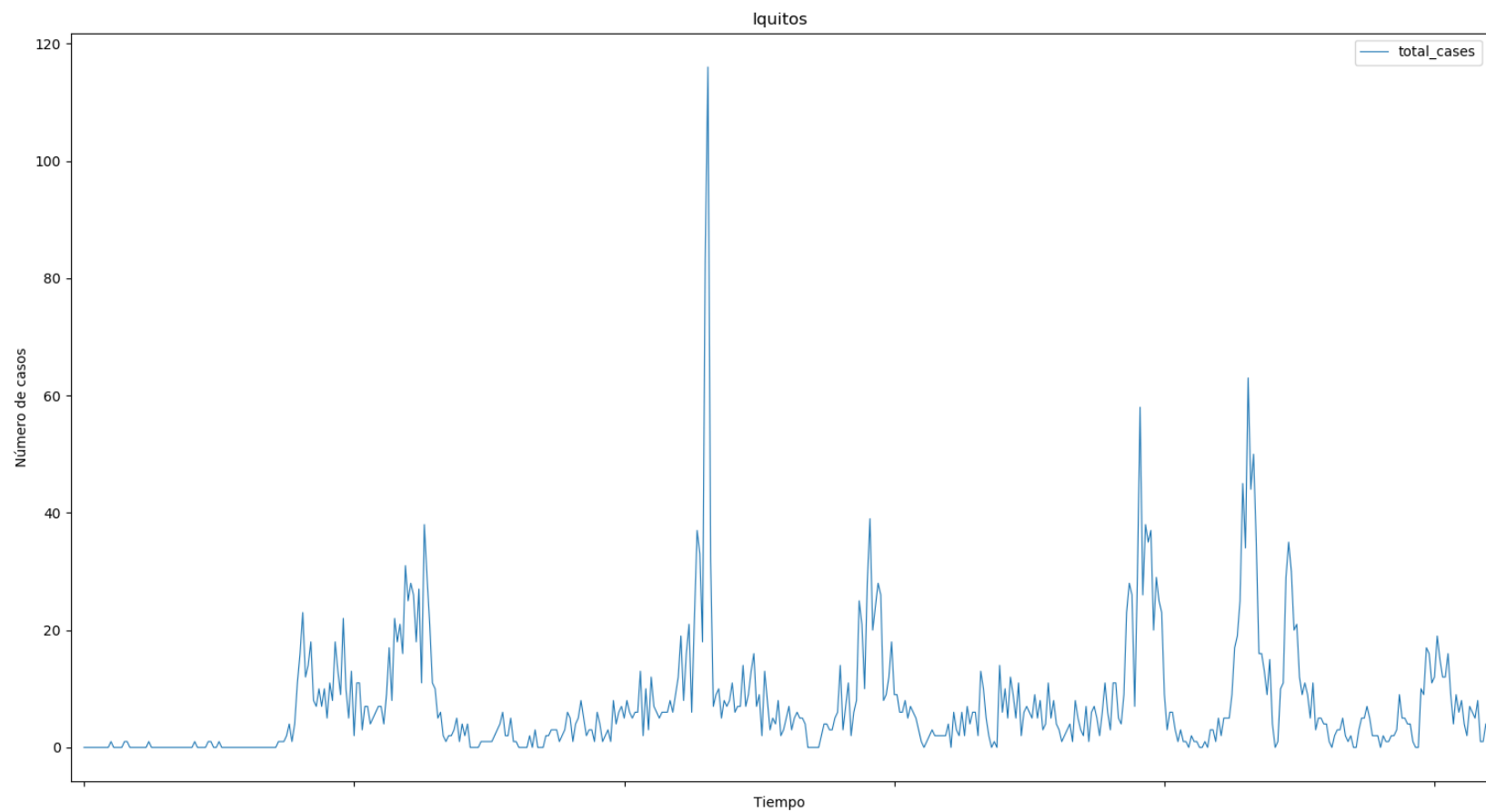
**Un modelo por ciudad**

**Considerar la temporalidad de los datos**

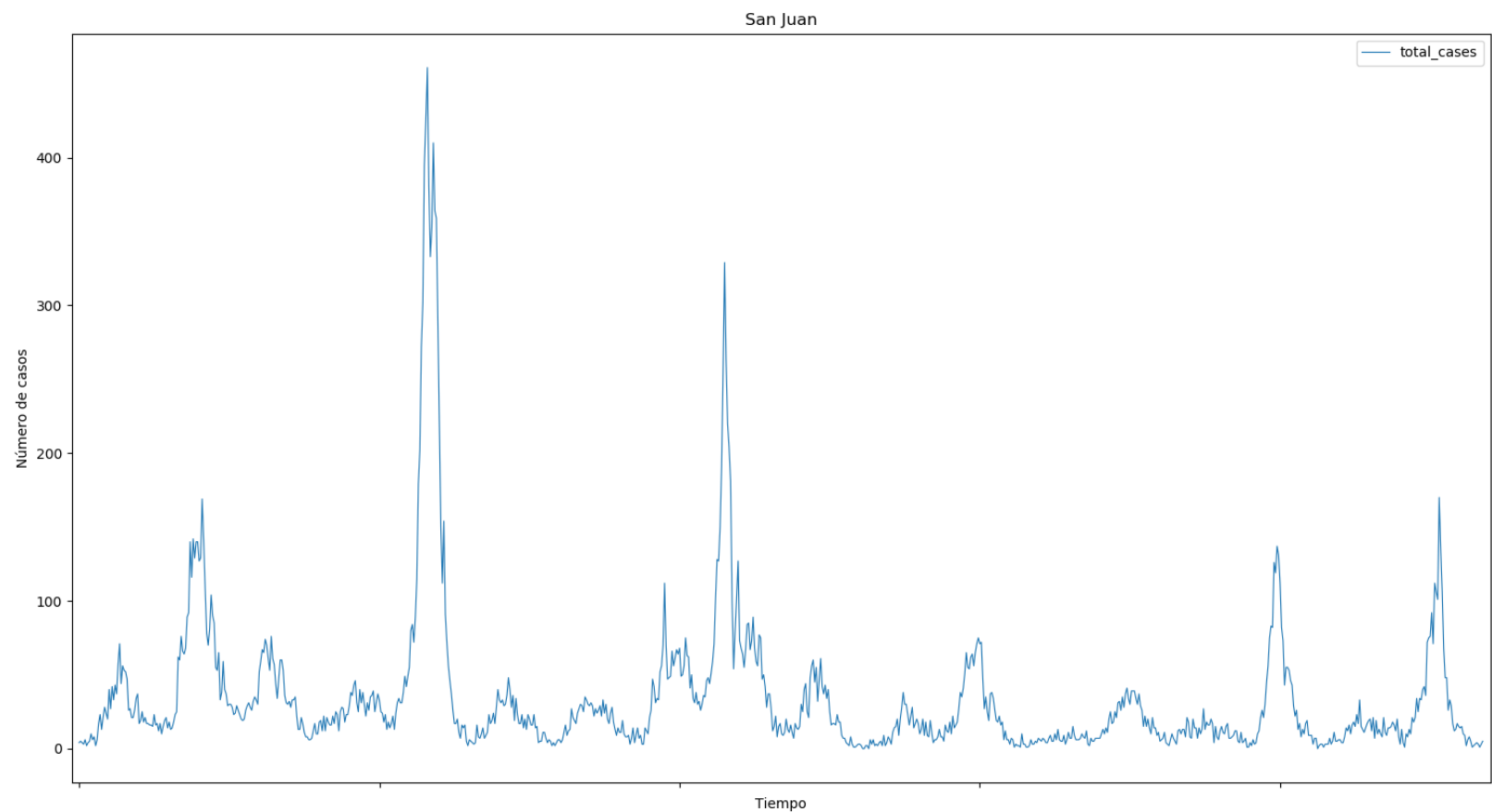
# Exploración de los datos

**<http://drivendata.co/blog/dengue-benchmark/>**

# Exploración de los datos



# Exploración de los datos



# Código - Ficheros

- **funciones.py**
- **validacion.py**
- **prediccion.py**
- **scriptLSTM.py**



# Mejor resultado – Selección de características

## Humedad

**reanalysis\_specific\_humidity\_g\_per\_kg**

**reanalysis\_dew\_point\_temp\_k**

## Temperatura

**station\_avg\_temp\_c**

**station\_min\_temp\_c**

## Tiempo

**weekofyear**

# Mejor resultado – Topología de la red

## 5 neuronas de entrada

El número de características seleccionadas

## Forma de cono o embudo

Más neuronas en las primeras capas

Menos al final

## Una neurona de salida

Solo se busca un valor

# Mejor resultado - Entrenamiento

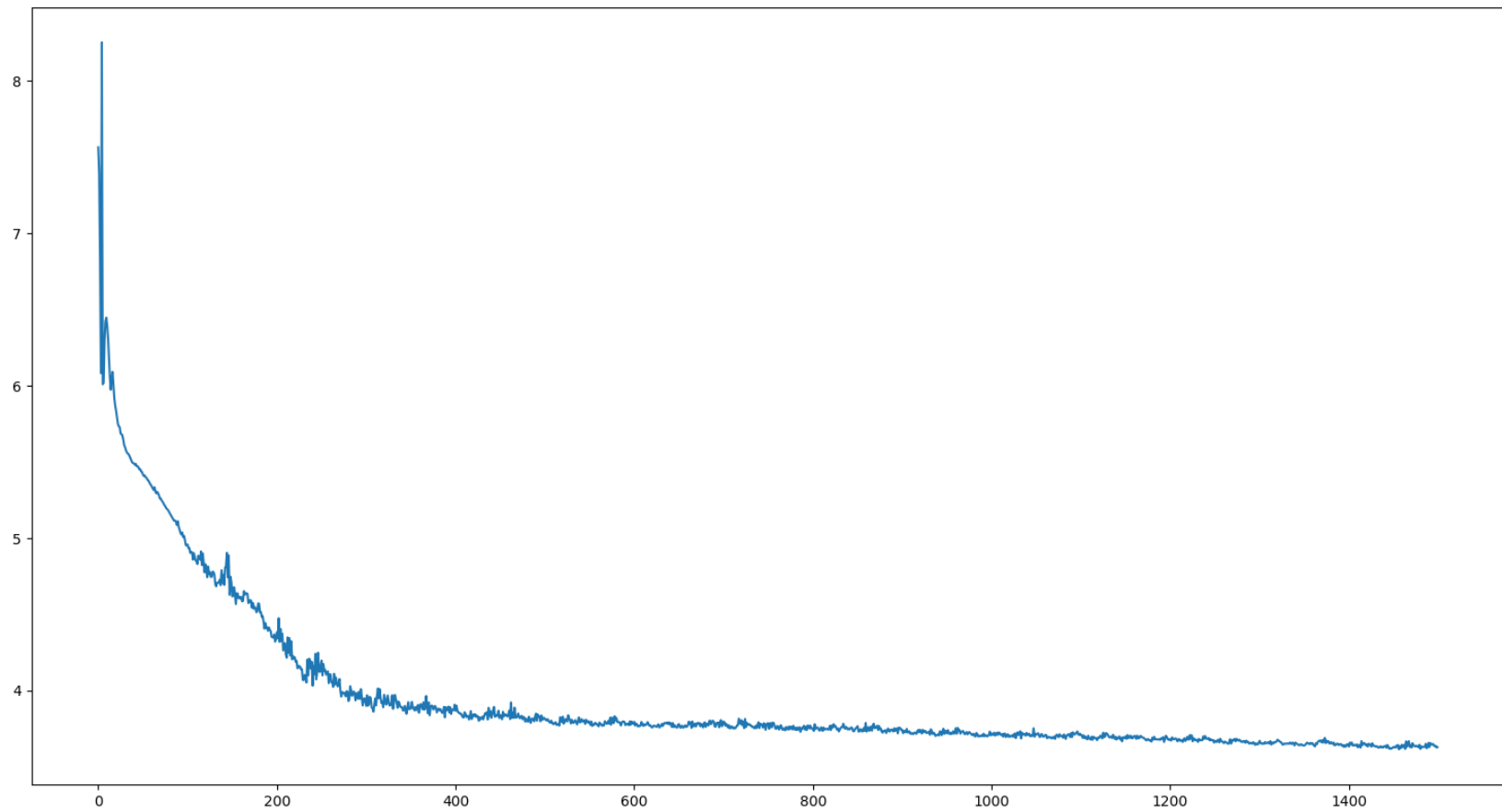
**Épocas: 1500**

**Algoritmo de optimización: Adam**

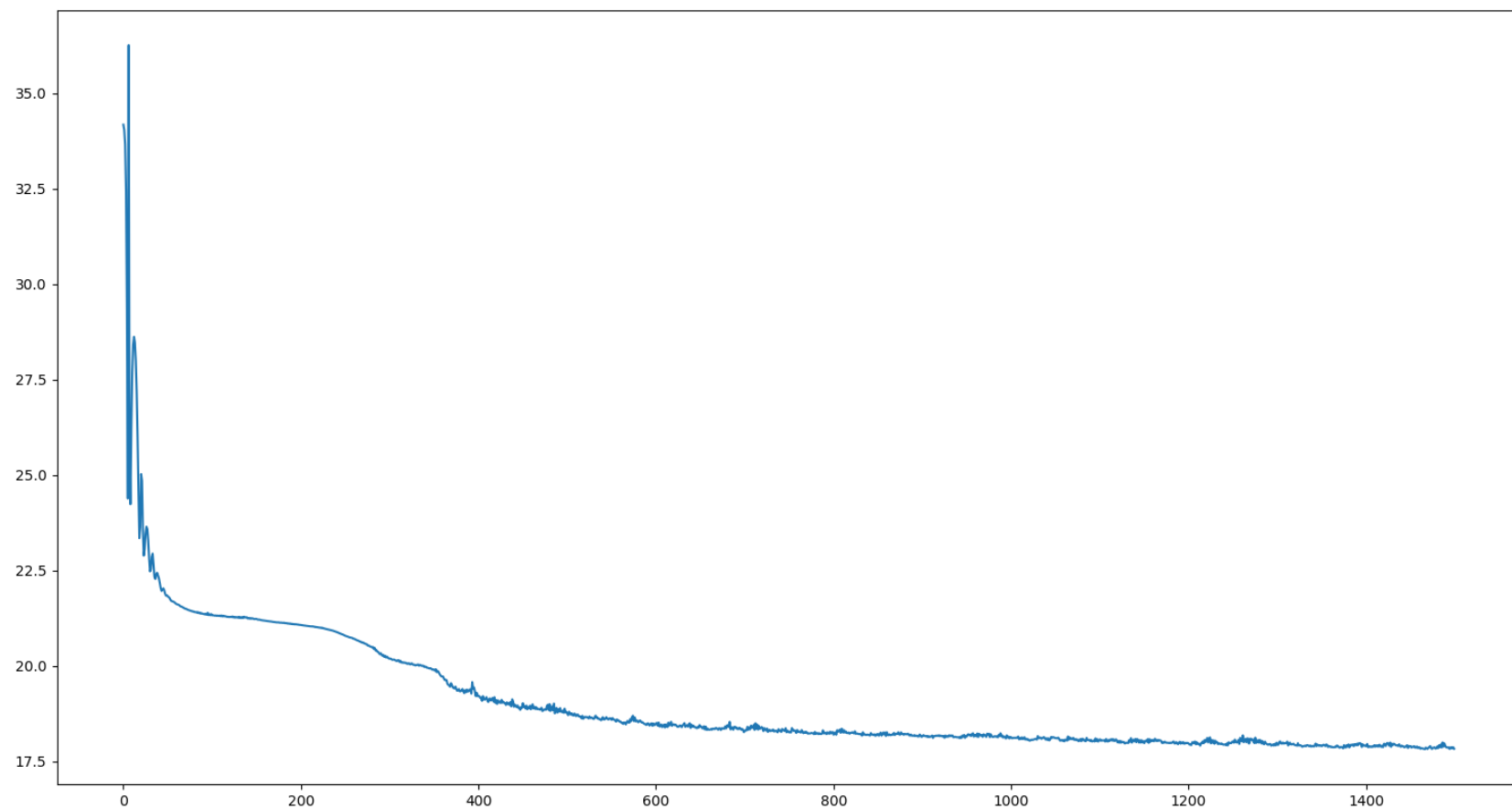
**Ratio de aprendizaje: 0.05**

**Ratio de decaimiento: 0.005**

# Entrenamiento Iquitos



# Entrenamiento San Juan



# Aproximaciones fallidas

**Eliminar filas con datos faltantes**

**Considerar el carácter temporal de los datos**

Suavizado exponencial

Suavizado promediando ventanas

RNN

**Suavizado temporal del número de casos**

# Problemas encontrados

**Convergencia hacia mínimos locales**

**Falta de convergencia**

**Número limitado de intentos**

# Lecciones aprendidas

**Trabajar con Keras y TensorFlow**

**Validación cruzada**

**Probar varias perspectivas**