



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

Jellyfish Forecast
Predicción de afluencia de medusas



Presentado por Pablo Santidrián Tudanca
en Universidad de Burgos — 20 de junio
de 2020

Tutor: José Francisco Díez Pastor y Álgvar
Arnaiz González



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. José Francisco Díez Pastor y D. Álar Arnaiz González, profesores del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Pablo Santidrian Tudanca, con DNI 71362353T, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Jellyfish Forecast.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 20 de junio de 2020

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. José Francisco Díez Pastor

D. Álar Arnaiz González

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general	III
Índice de figuras	v
Índice de tablas	vi
Introducción	1
Objetivos del proyecto	3
2.1. Objetivos generales	3
2.2. Objetivos técnicos	3
2.3. Objetivos personales	4
Conceptos teóricos	5
3.1. Medusas	5
3.2. Knowledge Discovery in Databases (KDD)	7
Técnicas y herramientas	13
4.1. Gestión del proyecto	13
4.2. Herramientas	14
4.3. Documentación	16
4.4. Bibliotecas	17
4.5. Bootstrap	18
Aspectos relevantes del desarrollo del proyecto	19
5.1. Recogida de datos	19
5.2. Ejecución remota	19
5.3. Preparación de los datos	20

5.4. Despliegue de la aplicación web	22
Trabajos relacionados	23
Conclusiones y Líneas de trabajo futuras	25
Bibliografía	27

Índice de figuras

3.1. Fases del proceso de reproducción de las medusas	6
3.2. Fases del proceso de KDD	8
3.3. Descomposición de una serie temporal	10
3.4. Ejemplo de series temporales no estacionarias	11
3.5. Ejemplo de serie temporal estacionaria	11
5.6. Estructura de datos resultante.	21
5.7. Croquis de referencia sobre los cuadrantes y el nombre de las variables asociadas.	21

Índice de tablas

Introducción

El cambio climático está provocando que los fenómenos naturales extremos sean cada vez más habituales. Esto afecta a distintas poblaciones locales como pueden ser la de las medusas.

Las medusas tienen periodos de aparición estacionales y se alimentan de plancton, por lo que su densidad es mayor en zonas donde este abunda. Estas zonas suelen ser lugares cercanos al talud continental donde además se reproducen [4].

La aparición de medusas cerca de las costas, es un fenómeno que se da cada vez con mayor frecuencia. Estas floraciones tiene efectos perjudiciales en ámbitos como el turismo o la pesca, así como los daños que pueden provocar a la salud de las personas llegando en algunos casos a causar enfermedades graves [25, 28].

Alguno de los factores que están provocando el aumento de los acercamientos de las medusas a las playas son [4, 21]:

- La **climatología**, influye principalmente el cambio climático con el descenso del nivel de lluvias y el aumento de las temperaturas, que favorecen el aumento de la salinidad y de la temperatura del agua.
- La **contaminación** provocada por la modificación de las zonas costeras o los vertidos cercanos a las costas provocan la proliferación de bacterias o plancton que sirve de alimento para las medusas.
- La **sobrepesca** causa un descenso de depredadores así como de otras especies con las que las medusas compiten por el alimento.

Con este proyecto se pretende generar un modelo con el que predecir el comportamiento de las poblaciones de medusas en las costas de Chile en

función de datos meteorológicos y marítimos obtenidos a través del programa europeo *Copernicus*¹.

La zona de estudio del proyecto se centra en Chile, pues son los datos de los que se dispone gracias a los colaboradores de la Universidad.

Un modelo es el resultado de aplicar un algoritmo a un conjunto de datos. El producto obtenido, es un conjunto reglas y patrones que pueden ser aplicados a otro grupos de datos con los que conseguir predicciones [26].

Para conseguir este modelo se van a aplicar técnicas de minería de datos y *machine learning*. Estas, se pueden entender como procesos para el descubrimiento de patrones y relaciones entre los diferentes datos de manera que nos permitan realizar predicciones a partir de los mismos.

Con el objetivo de poder aplicar estos algoritmos de la manera más eficaz posible, los datos son pre-procesados, es decir se eliminará la información (variables) que no es útil y se delimitará la zona geográfica de estudio.

¹Programa de observación terrestre de la Unión Europea. <https://marine.copernicus.eu/>

Objetivos del proyecto

A continuación, se detallarán los objetivos que han motivado la realización de este proyecto así como los resultados que se desean conseguir.

2.1. Objetivos generales

- Recopilar y filtrar los datos necesarios para entrenar un modelo predictivo.
- Utilizar técnicas de aprendizaje automático para predecir la llegada de medusas a las costas.
- Desarrollo de una aplicación web que permitía la consulta de las predicciones a los usuarios.

2.2. Objetivos técnicos

- Generar documentación en \LaTeX , aprendiendo dicho lenguaje de marcado para la edición de textos con acabado profesional.
- Utilizar un sistema de control de versiones con la plataforma GitHub junto a la extensión ZenHub para facilitar la gestión del proyecto.
- Generar scripts para recopilar y filtrar los datos necesarios para la realización del proyecto.
- Generar una estructura de datos sobre la que se obtendrán los modelos, utilizando los datos de avistamientos de medusas y los datos oceánicos obtenidos de *Copernicus*.

- Comparar los resultados de los diferentes modelos obtenidos.
- Realizar una web en la que mostrar los resultados del modelo de una manera fácil e intuitiva.

2.3. Objetivos personales

- Investigar diferentes técnicas y herramientas utilizadas para la minería de datos.
- Adquirir conocimientos sobre el desarrollo web.
- Aprender a generar documentación en L^AT_EX.

Conceptos teóricos

3.1. Medusas

Las medusas son animales marinos formados por un cuerpo gelatinoso del que cuelga un manubrio tubular, encontrando la boca en la parte inferior de este. Algunas especies, tienen tentáculos con celular urticantes denominadas cnidocitos. Las medusas se desplazan mediante contracciones de su cuerpo absorbiendo agua para luego ser expulsada de manera brusca provocando el movimiento [6].

Su reproducción es asexual, siendo esta, una fecundación externa mediante óvulos y espermatozoides que son liberados por los machos y las hembras respectivamente. Esto da lugar a la fecundación de gametos que se convertirán en larvas denominadas plánulas. Más adelante estas larvas se adhieren a alguna superficie donde se transformarán en pólipos para finalmente desprenderse la medusa adulta [16].

Su alimentación se basa principalmente en plancton aunque también son capaces de comer crustáceos, huevos o peces pequeños.

Comportamiento de las medusas

Los colonias de medusas están muy influenciadas por las condiciones climatológicas y marítimas. La temperatura, salinidad, el viento y las corrientes son los principales factores a tener en cuenta.

El peso de estos factores en los desplazamientos de las colonias varía en función de la etapa de desarrollo en la que se encuentren. Las que tienen un tamaño pequeño o mediano, están más condicionadas a la dirección de los vientos y las corrientes ya que, debido a su pequeño tamaño, no son capaces

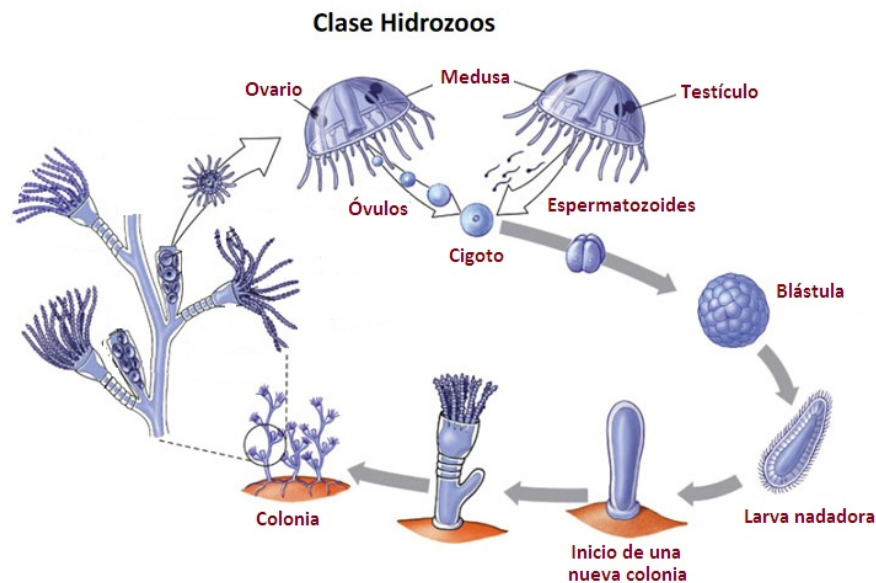


Figura 3.1: Fases del proceso de reproducción de las medusas [7].

de contrarrestar estas fuerzas. Por otro lado, en las medusas de un tamaño superior, estos factores tienen menos relevancia mientras que la salinidad del agua y la temperatura de la misma adquieren un mayor protagonismo. Hasta unos 25 grados el número de medusas va en aumento. A partir de ese punto, la concentración de las mismas decrece. *comentar la especie así como las características de la misma.*

“La reproducción de estas también se ve influenciada por la temperatura del agua pues según diferentes experimentos se ha demostrado que existe una relación entre el aumento de las temperaturas y una mayor reproducción asexual en varias especies gelatinosas” [23].

Los factores humanos también tienen su influencia en los movimientos de las poblaciones de medusas y en su reproducción. El aumento de materia orgánica provocado por vertidos como podrían ser los de una EDAR (Estación Depuradora de Aguas Residuales) o de una explotación agrícola, que provoca la eutrofización del medio haciendo que las medusas puedan desarrollarse de una manera más rápida. Del mismo modo, la construcción de estructuras costeras, proporcionan lugares donde pueden proliferar con mayor facilidad.

Teniendo en cuenta todo esto, diferentes estudios concluyen que estas alteraciones aleatorias del medio, tiene poca influencia en el desarrollo de las colonias de medusas, mientras que las condiciones ambientales que se repiten

anualmente tiene una mayor importancia en comparación. Esto remarca la importancia de un análisis de las condiciones ambientales que provocan estos brotes para poder anticiparse a ellos [21].

3.2. Knowledge Discovery in Databases (KDD)

Actualmente se recopila una gran cantidad de información de todos los ámbitos y es necesario darla un uso práctico. La **minería de datos** es un campo de las ciencias de la computación por el cual se tratan de descubrir nuevos patrones o relaciones en conjuntos de datos de manera automática. Con estas nuevas relaciones se trata de explicar comportamientos actuales o predecir resultados futuros [24].

Sin embargo, la minería de datos es solo una fase del proceso de descubrimiento del conocimiento (KDD). El *Knowledge Discovery* se trata de la búsqueda de conocimiento en los datos. Su objetivo es la extracción de información útil y previamente desconocida de conjuntos de datos. Estos los podemos obtener de multitud de formas diferentes por lo que es necesario un proceso de preparación de los mismos.

Posteriormente, es cuando podremos utilizar algoritmos de clasificación o regresión para lograr un modelo resultante en forma de reglas o patrones, que siendo interpretados, obtendremos nuevo conocimiento.citar

Un resumen de todo el proceso del KDD se puede observar en la Figura 3.2.

Añadir subsección de la obtención de los datos

Preprocesamiento de los datos

Se trata del primer paso una vez obtenido el conjunto de datos a utilizar. Estos datos en bruto no pueden ser utilizados en los algoritmos de minería sin un tratamiento previo. El conjunto de datos inicial debe ser transformado a un formato estándar para poder ser analizado.

Una vez conseguido, no se puede asumir que la información que contiene sea válida o correcta. Pueden existir variables que introduzcan ruido en la muestra o campos que se hayan introducido de manera errónea. También puede ocurrir que no se necesite la totalidad de la información contenida en el dataSet, sino que únicamente se requiera parte de la misma. Como ocurre

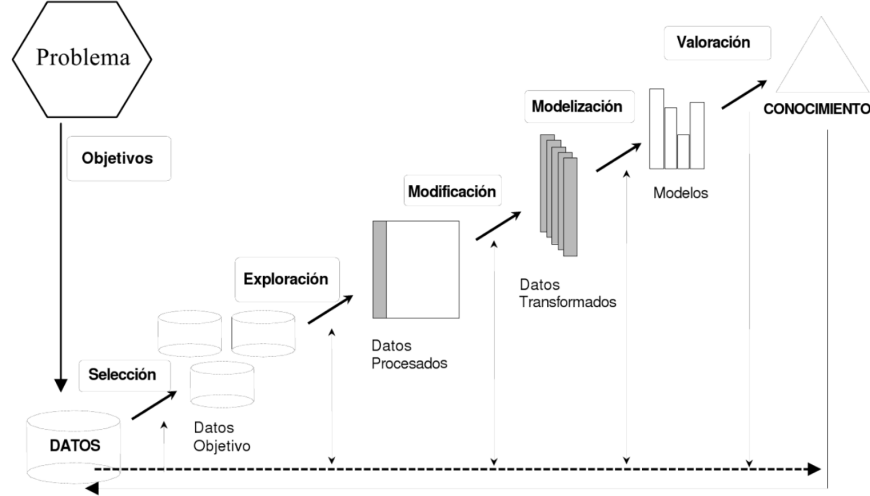


Figura 3.2: Fases del proceso de KDD [24].

en este caso, solo es de interés la el área geográfica de Chile y el área de los océanos cercana a sus costas.

Otro problema que se encuentra, son los valores nulos o la falta de los mismos. Ya sea por fallos es el grabado de la información o falta de la misma, es necesario corregir esos valores vacíos pudiendo descartar esas instancias o reemplazarlas por otros valores razonables teniendo en cuenta su contexto [22].

Finalmente se pueden aplicar al conjunto otras técnicas como la normalización de los datos.

Minería de datos

Como se ha comentado anteriormente, la minería de datos se trata de obtener patrones o relaciones a partir de conjuntos de datos.

En nuestro caso, se pretende conseguir un modelo capaz de encontrar relaciones entre el estado de la mar y la aparición de medusas en la costa. Para ello el modelo debe ser capaz de aprender a partir de los datos de entrada. Se pueden definir dos tipos de aprendizaje principales: supervisado y no supervisado.

Aprendizaje supervisado

Se trata de modelo que se elaboran a partir de conjuntos de datos etiquetados, es decir cada instancia tiene un atributo que es el valor a predecir. Este atributo será una clase ([mucho/poco], [alto/bajo]...) si se trata de un problema de clasificación, mientras que si se trata de un problema de regresión será un valor numérico.

Una vez entrenado el modelo, éste podrá etiquetar o predecir el valor de para nuevas instancias. En este caso tenemos un problema de regresión pues se intenta entrenar un modelo que prediga el número de avistamientos de medusas.

Aprendizaje no supervisado

En este caso, el entrenamiento del modelo no se hace con datos etiquetados sino que se les asignan características. A partir de los datos de entrada, se agrupa a los datos en conjuntos en función de estas características. Estos grupos se denominan *clusters* de instancias.

Series temporales

Las series temporales son sucesiones de datos que han sido tomados o medidos secuencialmente a lo largo del tiempo [27]. Estas tienen una serie de características:

- El valor medido o magnitud.
- La periodicidad, se trata de la repetición en el tiempo de los valores de manera cíclica.
- La tendencia es la variación que se observa en la media de los valores a largo plazo.
- El ruido que son valores aleatorios sin explicación.

Con estos componentes, podríamos separar una serie en sus diferentes partes (Figura 3.3).

Las series temporales pueden ser regulares o irregulares, es decir, serán regulares cuando la separación (distancia temporal) entre las diferentes medidas es constante, mientras que la irregulares esta separación es variable.

Un punto importante es el de la estacionariedad. Una serie temporal puede ser estacionaria cuando su media y varianza se mantienen constantes

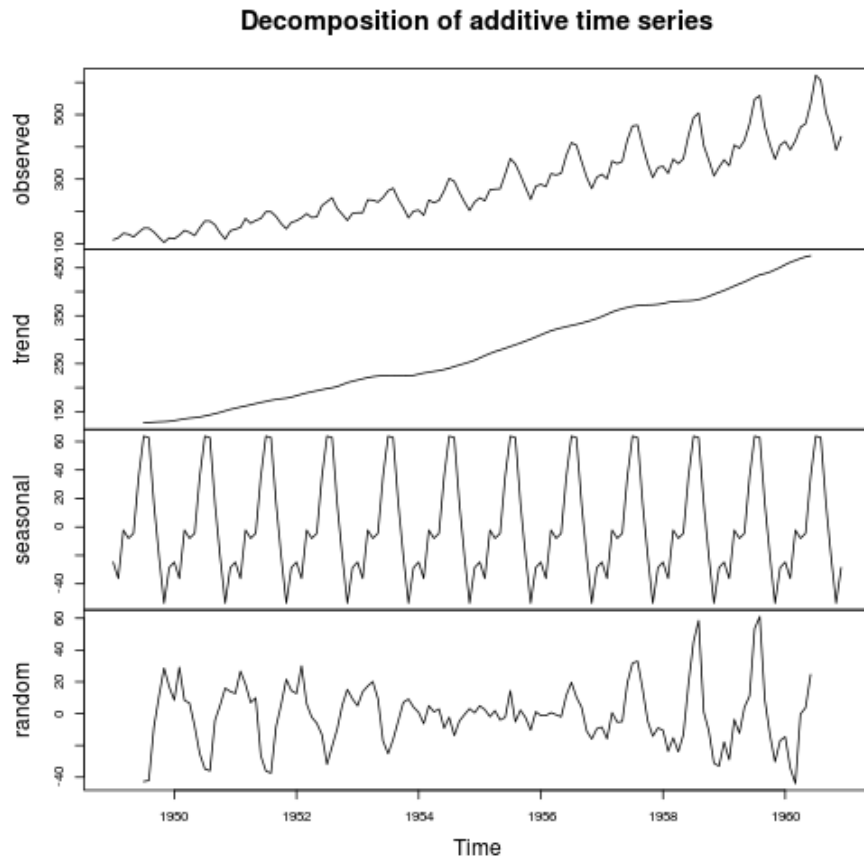


Figura 3.3: Descomposición de una serie temporal [8].

con el paso del tiempo, es decir, no dependen de este y además no presentan tendencia. Por otro lado, en una serie no estacionaria, estos valores sí cambian a lo largo del tiempo pudiendo encontrar variaciones de tendencia, varianza o co-varianza.

Tres ejemplos de esto podrían ser los reflejados en la figura 3.4.

- En el primer gráfico podemos observar como la media aumenta con el transcurso del tiempo, esto provoca una tendencia creciente, por lo que estamos frente a una serie no estacionaria.
- En el segundo caso la varianza de la serie no es constante por lo que tampoco es una serie estacionaria.

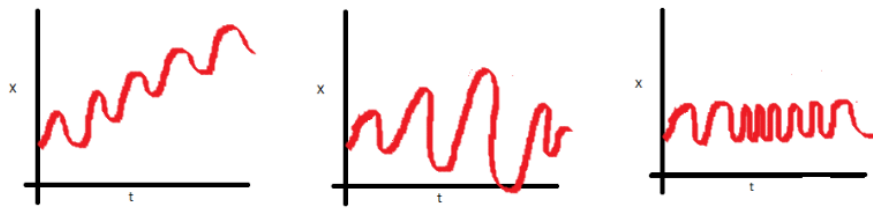


Figura 3.4: Ejemplo de series temporales no estacionarias [17].

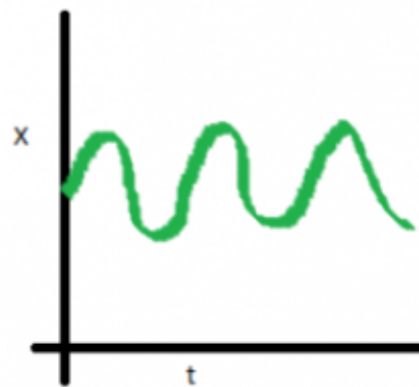


Figura 3.5: Ejemplo de serie temporal estacionaria [17].

- Por último caso, la co-varianza varía en función del tiempo por lo que no es una serie estacionaria.

En el caso de la figura 3.5, encontramos una serie con media, varianza y co-varianza constante por lo que si que podríamos hablar de una serie estacionaria.

Evaluación del modelo

Técnicas y herramientas

En este apartado se plasman las diferentes técnicas metodológicas y herramientas de desarrollo que se han utilizado en la realización del proyecto, así como las posibles alternativas que se han tenido en cuenta y el motivo de haberlas desechado.

4.1. Gestión del proyecto

Scrum

Scrum se trata de un marco de trabajo ágil destinado al manejo de proyectos de desarrollo *software*. Está destinado para equipos pequeños dividiendo el trabajo en objetivos que se van logrando de manera incremental a través de iteraciones denominadas *sprints* [29].

Git

Git se trata de un sistema de control de versiones gratuito y de código abierto para el manejo de proyectos. Se trata de un software de control de versiones de manera que se puede llevar un registro de cambios en los archivos y posibilita la coordinación de trabajos entre diferentes personas [3, 19].

GitHub

GitHub es una plataforma destinada a alojar proyectos, que se basa en el software de control de versiones Git. Esta plataforma utiliza un interfaz web desde la que se nos permite realizar control de código, documentación,

gestión de tareas y otros muchas funcionalidades además de integración con otros servicios. GitHub es gratuito para proyectos *open source* [20, 12].

Alternativas

Otras alternativas a GitHub fueron, Gitlab y Bitbucket. Ambos servicios son bastante similares a GitHub en funcionalidades y basados en Git.

- Bitbucket fue rechazado rápidamente por la falta de familiaridad en su uso ya que no lo había usado nunca, únicamente había visto repositorios en la web.
- Gitlab es un entorno más conocido, ya que es el software que he utilizado en las prácticas de empresa para el control de código dentro del equipo de trabajo del que formo parte.

Finalmente se decidió usar GitHub por haber sido utilizado en clase de gestión de proyectos por lo que se tenía un mayor conocimiento de su funcionamiento, así como por su integración con ZenHub del que se hablará a continuación.

ZenHub

ZenHub es una herramienta para gestión de proyectos que se integra con GitHub. Este complemento añade a GitHub un tablero canvas en el cual se representan las *issues*. Es posible estimar tareas, así como darlas prioridades y visualizar gráficos como el gráfico *burndown* [14].

Alternativas

Se plantearon otras alternativas como Trello o GitHub projects, pero finalmente se eligió ZenHub por su facilidad de uso y el haber sido usado anteriormente, requisito que las otras alternativas no cumplían.

4.2. Herramientas

Entorno de desarrollo de Python

Para el desarrollo en Python se eligió Visual Studio Code. Esta es una aplicación totalmente gratuita basada en el framework Electron y posee gran cantidad de extensiones para facilitar la tarea a la hora de la programación

como auto completado con IntelliSense. Además tiene integración con Git para el control de versiones [11, 10].

Alternativas

Se estudiaron otras alternativas como PyCharm o Jupyter Notebook pero finalmente se eligió Visual Studio Code por la familiaridad con la misma y gran compatibilidad con diferentes lenguajes.

Jupyter Notebook

También se ha utilizado Jupyter Noteebok para el desarrollo con Python. Esta herramienta se ha utilizado para realizar todas las pruebas debido a su facilidad para documentar el código y la segmentación del mismo en apartados separados.

tmux

Se trata de un herramienta que permite lanzar múltiples terminales en una misma ventana, cada uno de manera independiente y corriendo en segundo plano. Esta herramienta se ha utilizado para la descarga de los datos obtenidos de *Copernicus*. En un principio se utilizó únicamente una conexión ssh con un equipo de cómputo de Universidad. Esto daba varios errores, por una parte, la conexión se cerraba perdiendo el proceso de descarga. También era necesario que un equipo personal estuviese encendido constantemente durante la descarga para que no se detuviera la conexión. Para solucionar estos problemas se abrió una sesión de tmux corriendo en segundo plano, pudiendo así cerrar a conexión ssh y abrirla solo para consultar el proceso. **Añadir si se usa para la ejecucion del modelo.**

Pencil

Software gratuito para la realización de prototipos de interfaces gráficas. Permite la instalación de paquetes para crear maquetas de múltiples plataformas.

Creately

Aplicación web utilizada en la fase de modelado para la creación de los diagramas de uso. Dispone de una versión gratuita con opciones reducidas.

Heroku

Se trata de una aplicación de hosting para aplicaciones web en diferentes lenguajes, en mi caso, Python era el que me interesaba. Tiene limitaciones en su versión gratuita como un límite de tamaño en las bases de datos o un máximo de horas mensuales pero esto no nos afecta en la actualidad.

Alternativas

Se investigaron otras alternativas como PythonAnywhere o DigitalOcean pero por motivos de precios o no ofrecer el autodespliegue desde Github, finalmente se optó por Heroku.

4.3. Documentación

L^AT_EX

L^AT_EX es un sistema de composición de texto plano destinado a la composición de textos con una calidad tipográfica alta. Incluye características diseñadas para la elaboración de documentación técnica y científica siendo un estándar en la publicación de documentos de investigación. L^AT_EX es totalmente gratuito [5].

Alternativas

Otras opciones valoradas fueron Microsoft Word y OpenOffice. La Universidad proporciona una plantilla para L^AT_EX y OpenOffice por lo que Microsoft Word fue la primera descartada. Finalmente debido al enfoque más técnico que proporciona L^AT_EX frente a OpenOffice, se decidió utilizarlo.

T_EXstudio

T_EXstudio se trata de un editor de textos gratuito, que ofrece diversas herramientas para elaborar documentación con L^AT_EX haciendo la escritura más confortable e intuitiva [9].

Alternativas

Las alternativas a este editor que se estudiaron fueron T_EXmaker y Overleaf. Se terminó eligiendo T_EXstudio por ser un editor instalado en local, permitiendo el trabajo aunque no se tuviera conexión a Internet así

como por ser más intuitivo y fácil de utilizar que \TeX maker.

Zotero

La herramienta Zotero es un software gratuito para la gestión de referencias pudiendo recoger, organizar y citar creando referencias bibliográficas para cualquier editor. También cuenta con integración en el navegador. Una vez recopilados todas las citas, se puede exportar a un fichero Bib \TeX para la utilizarse con \LaTeX [15].

4.4. Bibliotecas

Flask

Flask es un framework ligero para el desarrollo de aplicaciones web en Python bajo el modelo Modelo-Vista-Controlador (MVC). Está diseñado para desarrollar aplicaciones de manera rápida y sencilla y con capacidad de escalar a aplicaciones más complejas [2].

Xarray

Xarray se trata de un proyecto de código abierto desarrollado para Python que facilita el trabajo con matrices multidimensionales etiquetadas utilizando la librería NumPy. Consta de una gran variedad de funciones para el análisis y la visualización de estructuras de datos. Está inspirado en el funcionamiento de la librería pandas y diseñado para funcionar con archivos de tipo netCDF [13].

Pandas

Esta biblioteca se trata de una extensión de NumPy y está destinada a la manipulación y análisis de datos en lenguaje Python. Permite trabajar con estructuras de datos y operaciones para su transformación pudiendo estas ser tablas temporales o series numéricas [18].

netCDF

[Añadir descripción.](#)

FtpLib

Biblioteca destinada a la implementación de la parte del cliente en el protocolo FTP. Desarrollada para el lenguaje Python, nos permite automatizar accesos a servidores FTP [\[1\]](#).

tqdm

Pequeña librería utilizada para mostrar una barra de progreso a la hora de realizar la descarga de los datos del FTP.

Folium

Se trata de una biblioteca que nos permite la visualización de mapas, pudiendo superponer elementos en los mismos. Folium utiliza a su vez una biblioteca de javaScript llamada *leaflet*.

4.5. Bootstrap

Bootstrap es un conjunto de herramientas para facilitar el desarrollo en HTML, CSS y JavaScript.

Aspectos relevantes del desarrollo del proyecto

5.1. Recogida de datos

La obtención de los datos necesarios para el posterior análisis, se realizó a través de la organización europea *Copernicus*. Esta cuenta con una serie de datos recopilados por satélites de todo el mundo.

Para la descarga de estos datos, se encontraban disponibles dos alternativas. Por un lado, podían ser descargados a través de un servidor FTP y por el otro, había disponible una API de reciente lanzamiento.

En un principio se priorizó la opción de la descarga a través de la API ya que podían descargarse los datos ya tratados reduciendo el trabajo previo al análisis. Finalmente se descartó esta idea por la lentitud de respuesta del servicio, así como diferentes errores que se producían, haciendo que la descarga de la totalidad de los datos no estuviese asegurada. Por este motivo se descargaron los datos a través del servidor FTP y posteriormente, eran tratados eliminando las variables y zonas geográficas que no eran necesarias.

Colocar enlace de donde se han sacado los datos así como una cuenta de acceso.

5.2. Ejecución remota

Para la ejecución de los scripts utilizados para la descarga de los datos, la ejecución del modelo y la realización de las diferentes pruebas, se ha

utilizado un equipo de cómputo de la Universidad mediante una conexión ssh y una VPN. Todo esto ha hecho que surjan diferentes problemas.

En primer lugar no se disponía de permisos de administrador para instalar las diferentes bibliotecas necesarias. Esto se solucionó mediante la instalación de la distribución local Anaconda que nos permite ser instalada para un único usuario. Con esto, se creó un entorno virtual en el que poder instalar todas las bibliotecas necesarias.

tmux

Como se ha explicado anteriormente al hablar de tmux (4.2), se ha tenido problemas a la hora de ejecutar los diferentes scripts en el equipo de cómputo.

Para la ejecución de las diferentes pruebas, se utilizó la conexión ssh lanzando un proceso de Jupyter Notebook sin interfaz gráfica mediante el comando `jupyter notebook --no-browser`. Posteriormente, se conecta el puerto en que se ejecuta jupyter en la máquina de cómputo, con un puerto en un equipo personal y de esta manera, trabajar como si la ejecución de los *Notebooks* se realizase en local.

Existían casos en los que la conexión ssh se cerraba o se caía la conexión VPN, por lo que se perdía el proceso de ejecución o los últimos cambios realizados en los notebooks no se guardaban. Para eso se utilizó la herramienta tmux permitiendo, que aunque la conexión se perdiera, los procesos en segundo plano no se perdían y así poder continuar con el trabajo.

5.3. Preparación de los datos

Tras haber descargado los datos oceánicos, disponemos de dos fuentes de datos. Por una lado, el estado de los océanos en las diferentes fechas y coordenadas, y por otro, un registro de avistamientos de medusas.

Los datos oceánicos están agrupados por cuadrantes separados cada 0.0833 grados (1/12). Por esto se redondearon las coordenadas de los avistamientos para coincidir con estos pasos.

A continuación se enlazan en un solo DataFrame ([Describir dataframe](#)) los avistamientos con la variables oceánicas recogidas de ese cuadrante y en la fecha del avistamiento quedando la siguiente estructura de datos como la que encontramos en la figura 5.3.

Latitud	Longitud	Fecha	Avistamientos	Profundidad	mlotst	zos	bottomT	thetao	so	uo	vo
-25.416667	-70.500000	2014-05-01	1	0.494025	12.5126	0.194708	12.5373	15.8861	34.7331	-0.0592059	-0.14771
				5.078224	12.5126	0.194708	12.5373	15.8524	34.7331	-0.0640889	-0.175176
				9.572997	12.5126	0.194708	12.5373	15.8209	34.7316	-0.0646992	-0.189215
-18.500000	-70.333333	2014-05-01	1	0.494025	10.5289	0.219428	15.3323	17.6198	34.8888	-0.0714133	0.136113
				5.078224	10.5289	0.219428	15.3323	17.2711	34.9101	-0.0622578	0.0860622

Figura 5.6: Estructura de datos resultante.

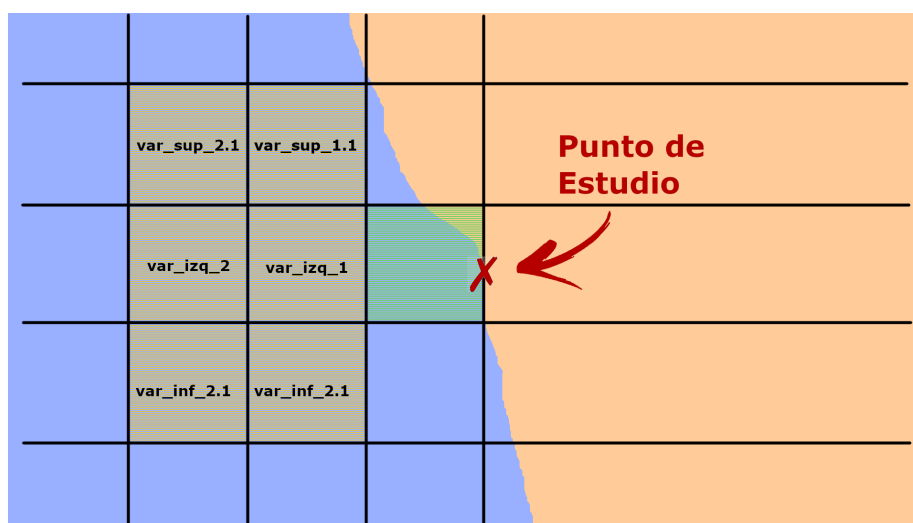


Figura 5.7: Croquis de referencia sobre los cuadrantes y el nombre de las variables asociadas.

Esta estructura inicial, contiene poca información, pues no se puede predecir con exactitud la aparición de medusas observando únicamente el cuadrante más próximo a las playas. Por ello se añadieron más lecturas de los cuadrantes adyacentes mar adentro. Estos nuevos cuadrantes fueron incluidos a la estructura inicial modificando el nombre con una etiqueta al final de las mismas para poder identificarlos. El resultado de como se organizan estas etiquetas se puede observar en la figura 5.3.

Tras unir todos los cuadrantes nos encontramos que ciertas celdas no contenían información. Esto se debe principalmente por dos causas: Las profundidades se toman a 0,5 y 10 metros, esto hace que en los cuadrantes más cercanos a las costas existiesen zonas con una profundidad inferior. También debido a la forma de la línea de costa, existen zonas en las que ya sean cuadrantes inferiores o superiores, están situados tierra adentro.

La solución que se buscó a este problema, fue la de utilizar los datos del cuadrante más cercano por la izquierda que tenga información válida.

CrossCorelation

5.4. Despliegue de la aplicación web

Al tratarse se una aplicación web, un aspecto importante es el despliegue de la misma en un servidor al que poder acceder desde cualquier lugar. Como se ha mencionado en la sección de herramientas (4.2), se ha utilizado la aplicación *Heroku*.

Esta nos ofrece hosting gratuito, además de desplegarse automáticamente cada vez de que se realiza un *push* en el repositorio, si este está asociado a la cuenta de heroku. Por este motivo, se tuvo que realizar un segundo repositorio en Github en el que incluir unicamente la parte de la aplicación web, ya que para poder realizar el autodespliegue los archivos deben estar en el directorio raíz de nuestro repositorio.

Se añadió una referencia desde el repositorio principal a este segundo ya que *Git* lo permite, además de incluir el enlace también en el archivo README. El repositorio se encuentra en <https://github.com/psnti/WebJellyfishForecast>.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] 20.8. ftplib — FTP protocol client — Python 2.7.17 documentation.
- [2] Flask. Library Catalog: palletsprojects.com.
- [3] Git.
- [4] Las proliferaciones de medusas. Library Catalog: www.miteco.gob.es.
- [5] LaTeX - A document preparation system.
- [6] Medusozoa. Page Version ID: 124010357.
- [7] Reproducción en hidrozoo. Library Catalog: Blogger.
- [8] Series temporales con R: Estacionariedad y raíces unitarias | Finanzas-Zone. Library Catalog: finanzaszone.com.
- [9] TeXstudio.
- [10] Visual Studio Code - Code Editing. Redefined. Library Catalog: code.visualstudio.com.
- [11] Visual Studio Code - Wikipedia, la enciclopedia libre.
- [12] The world's leading software development platform · GitHub.
- [13] xarray: N-D labeled arrays and datasets in Python — xarray 0.14.1 documentation.
- [14] ZenHub - Agile Project Management for GitHub. Library Catalog: www.zenhub.com.

- [15] Zotero | Your personal research assistant.
- [16] Reproducción de las medusas, sus etapas, pólipos y larvas plánulas., October 2016. Library Catalog: medusas.wiki Section: Información general.
- [17] An Introduction To Non Stationary Time Series In Python, September 2018. Library Catalog: www.analyticsvidhya.com.
- [18] Pandas, November 2019. Page Version ID: 121498623.
- [19] Git, February 2020. Page Version ID: 123779424.
- [20] GitHub, March 2020. Page Version ID: 124035823.
- [21] Lisandro Benedetti-Cecchi, Antonio Canepa, Veronica Fuentes, Laura Tamburello, Jennifer E. Purcell, Stefano Piraino, Jason Roberts, Ferdinando Boero, and Patrick Halpin. Deterministic Factors Overwhelm Stochastic Environmental Fluctuations as Drivers of Jellyfish Outbreaks. *PLOS ONE*, 10(10):e0141060, October 2015. Publisher: Public Library of Science.
- [22] Max Braner. *Principles of Data Mining*. Third edition edition.
- [23] Antonio Canepa, Verónica Fuentes, Mar Bosch-Belmar, Melissa Acevedo, Kilian Toledo-Guedes, Antonio Ortiz, Elia Durá, César Bordehore, and Josep-Maria Gili. Environmental factors influencing the spatio-temporal distribution of carybdea marsupialis (lineo, 1978, cubozoa) in south-western mediterranean coasts. 12(7):e0181611. Publisher: Public Library of Science.
- [24] Daniel Santín González César Pérez López. Minería de datos: técnicas y herramientas - César Pérez López - Google Libros.
- [25] Lisa-ann Gershwin, Anthony J. Richardson, Kenneth D. Winkel, Peter J. Fenner, John Lippmann, Russell Hore, Griselda Avila-Soria, David Brewer, Rudy J. Kloser, Andy Steven, and Scott Condie. Biology and ecology of Irukandji jellyfish (Cnidaria: Cubozoa). *Adv. Mar. Biol.*, 66:1–85, 2013.
- [26] Minewiskan. Modelos de minería de datos (analysis services, minería de datos) - SQL server 2014. Library Catalog: docs.microsoft.com.
- [27] Antonio Canepa Oneto, Mario Juez Gil, and Álgar Arnaiz González. Introducción al análisis de series temporales con R y Python. page 17.

- [28] James Tibballs, Ran Li, Heath A. Tibballs, Lisa-Ann Gershwin, and Ken D. Winkel. Australian carybdeid jellyfish causing "Irukandji syndrome". *Toxicon*, 59(6):617–625, May 2012.
- [29] Wikipedia contributors. Scrum (software development) — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Scrum_\(software_development\)&oldid=943108748](https://en.wikipedia.org/w/index.php?title=Scrum_(software_development)&oldid=943108748), 2020. [Online; accessed 10-March-2020].