

Explorando el conjunto de datos de Palmer's Penguins con una aplicación web de visualización interactiva

Ariel Santa Cruz Pacheco Espino
Departamento de Ciencias
Básicas Universidad Autónoma
de Aguascalientes

al344583@edu.uaa.mx

El presente trabajo explora técnicas de análisis de datos y visualización utilizando el popular conjunto de datos "Palmer Penguins" y la biblioteca de Python Streamlit. El conjunto de datos contiene mediciones de tres especies de pingüinos de la Antártida: Adelie, Gentoo y Chinstrap, incluyendo variables como longitud y profundidad del pico, longitud de la aleta y peso, así como información sobre la especie y la región de anidación.

Aplicaremos diversas técnicas de análisis, como estadística descriptiva, modelado predictivo y segmentación de grupos, con el fin de identificar patrones, diferencias significativas entre especies y regiones, y otros hallazgos relevantes. Además, desarrollaremos una aplicación web interactiva utilizando Streamlit, que permitirá visualizar y explorar los datos de manera dinámica e intuitiva mediante gráficos y herramientas de análisis integradas.

Palabras clave: Ciencia de datos, análisis de datos, visualización de datos, pingüinos, Streamlit, aplicación web interactiva, Python, estadística descriptiva, modelado predictivo, segmentación de grupos.

I. INTRODUCCIÓN

El análisis de datos y la visualización son herramientas esenciales en el campo de la ciencia de datos. Permiten a los investigadores y analistas obtener información valiosa de grandes conjuntos de datos y comunicar sus hallazgos de manera efectiva. En este proyecto, utilizaremos el popular conjunto de datos "Palmer Penguins" para explorar técnicas de análisis de datos y visualización en Python.

El conjunto de datos "Palmer Penguins" fue recopilado por el ecólogo Dr. Kristen Gorman y su equipo en la Isla Torgersen, cerca de la Base Palmer en la Antártida. Contiene mediciones de pingüinos de tres especies diferentes: Adelie, Gentoo y Chinstrap. Las variables incluyen medidas físicas como longitud del pico, profundidad del pico, longitud de la aleta y peso, así como información sobre la especie y la región de anidación.

Este conjunto de datos es valioso para la práctica de ciencia de datos porque permite realizar una variedad de análisis y visualizaciones. Podremos explorar patrones de tamaño y peso entre especies, identificar diferencias demográficas o geográficas, y crear visualizaciones interactivas para comunicar estos hallazgos de manera efectiva.

Una de las herramientas que utilizaremos en este proyecto es Streamlit, una biblioteca de código abierto de Python que permite crear aplicaciones web interactivas de ciencia de datos de forma sencilla y rápida. Streamlit facilita la visualización y el intercambio de datos, análisis y modelos de aprendizaje automático mediante una interfaz web atractiva y fácil de usar.

Utilizando el conjunto de datos "Palmer Penguins" y la biblioteca Streamlit, desarrollaremos una aplicación web interactiva que nos permitirá explorar y visualizar los datos de manera dinámica. Implementaremos técnicas de análisis estadístico, modelado predictivo y segmentación de grupos para obtener información valiosa sobre las características de los pingüinos y sus diferencias entre especies y regiones de anidación.

Este proyecto no solo nos permitirá poner en práctica nuestras habilidades de análisis de datos y visualización, sino también aprender sobre la creación de aplicaciones web interactivas utilizando Streamlit, una herramienta cada vez más popular en el campo de la ciencia de datos.

II. DESARROLLO

A. *MLOps*

En este proyecto de MLOps, aplicamos conocimientos fundamentales de desarrollo de aplicaciones de aprendizaje automático en la nube mediante la implementación del proceso MLOps (Nivel 0). Este proceso se centra en la integración del ciclo de vida de las aplicaciones de aprendizaje automático, cubriendo tres etapas esenciales: Ciencia de datos (ML), Desarrollo de aplicaciones y Operaciones de TI. La combinación de estas etapas permite una gestión integral y eficiente de los modelos de ML desde la concepción y el desarrollo hasta la implementación y el funcionamiento en entornos de producción en la nube.

Etapas de desarrollo

A continuación, se detallan las etapas de desarrollo específicas para este proyecto:

1. Desarrollo de ML: Implica experimentar y desarrollar un procedimiento de entrenamiento de modelos robusto y reproducible que consta de múltiples tareas, desde la preparación y transformación de datos hasta el entrenamiento y evaluación de modelos.
2. Operacionalización de la formación: Implica la automatización del proceso de empaquetado, prueba e implementación de canalizaciones de entrenamiento repetibles y confiables.
3. Formación continua: Se refiere a la ejecución repetida del proceso de entrenamiento en respuesta a nuevos cambios de datos o código, o de acuerdo con una programación, potencialmente con nuevas configuraciones de entrenamiento.
4. Implementación de modelos: Implica empaquetar, probar e implementar un modelo en un entorno de servicio para la experimentación en línea y el servicio de producción.
5. Servicio de predicción: Consiste en desplegar el modelo en producción para realizar inferencias. La supervisión continua implica la supervisión de la eficacia y la eficiencia de un modelo implementado.
6. Gestión de datos y modelos: Es una función central y transversal para gobernar los artefactos de ML para admitir la auditabilidad, el seguimiento y el cumplimiento. La administración de datos y modelos también puede promover la capacidad de compartir, reutilizar y descubrir activos de ML.

III. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO DE SISTEMAS

Para llevar a cabo el desarrollo del sistema, se utilizaron una serie de herramientas Python como lenguaje de

programación. Se empleó Python para entrenar el modelo de aprendizaje automático y desarrollar la funcionalidad del sistema. Además, se implementó Streamlit como una biblioteca para mostrar de forma interactiva los resultados obtenidos del modelo, proporcionando una experiencia fácil de usar.

En cuanto a infraestructura y despliegue, se eligió AWS (Amazon Web Services) como plataforma de despliegue y Amazon Lightsail como servidor privado virtual (VPS) para alojar el sistema de forma segura y escalable. Docker se utilizó para la gestión de contenedores, creando imágenes del sistema almacenadas en contenedores para su portabilidad e implementación en varios entornos. Visual Studio Code sirvió como entorno de desarrollo integrado (IDE) para la codificación eficiente y la colaboración en el desarrollo de proyectos. Kaggle fue fundamental como plataforma desde la cual se obtuvo el conjunto de datos necesario para el entrenamiento y la validación del modelo. Por último, se utilizó GitHub como plataforma de desarrollo colaborativo para alojar y gestionar el código fuente del proyecto, lo que permitió un flujo de trabajo colaborativo y un control de versiones eficiente.

A. *Amazon Web Services y Amazon Lightsail*

Amazon Web Services (AWS) es una plataforma de servicios en la nube líder que ofrece una amplia gama de herramientas y servicios para alojar y administrar aplicaciones, almacenar datos, procesar análisis y más. En este proyecto, utilizamos Amazon Lightsail, un servicio de AWS que simplifica la implementación y gestión de recursos informáticos en la nube.

Lightsail nos permitió crear y administrar instancias de máquinas virtuales, contenedores y bases de datos de manera sencilla y escalable. Esto nos permitió alojar nuestra aplicación web interactiva de ciencia de datos en la nube, haciéndola accesible a través de Internet.

B. *Python como lenguaje de programación*

Python es un lenguaje de programación ampliamente utilizado en diversos campos, incluida la ciencia de datos y el aprendizaje automático. En este proyecto, utilizamos Python y sus poderosas bibliotecas, como Pandas, NumPy y Scikit-learn, para realizar el análisis exploratorio de datos, el modelado predictivo y la segmentación de grupos.

Python es conocido por su sintaxis limpia y legible, lo que facilita el desarrollo y el mantenimiento del código. Además, su gran comunidad y amplia documentación lo convierten en una excelente opción para proyectos de ciencia de datos.

C. *Streamlit para la visualización de datos*

Streamlit es una biblioteca de código abierto de Python que permite crear aplicaciones web interactivas de ciencia de datos de forma sencilla y rápida. En este proyecto, utilizamos Streamlit para desarrollar una aplicación web que visualiza y

explora de manera dinámica el conjunto de datos "Palmer Penguins".

Streamlit facilita la creación de interfaces de usuario atractivas y personalizables, lo que nos permitió presentar los datos, las visualizaciones y los resultados del análisis de una manera intuitiva y accesible para los usuarios.

D. Docker como gestión de contenedores

Docker es una plataforma de software que permite crear, implementar y ejecutar aplicaciones dentro de contenedores. En este proyecto, utilizamos Docker para empaquetar nuestra aplicación web de ciencia de datos y todas sus dependencias en un contenedor portable y consistente.

Docker nos permitió garantizar que nuestra aplicación se ejecutara de manera coherente en diferentes entornos, como nuestras máquinas locales y la instancia de Amazon Lightsail en la nube. Además, facilitó el proceso de implementación y escalado de la aplicación.

E. Visual Studio Code como IDE

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. En este proyecto, utilizamos VS Code como nuestro entorno de desarrollo integrado (IDE) para escribir, editar y depurar nuestro código Python.

VS Code ofrece una amplia gama de características y extensiones que mejoran la experiencia de desarrollo, como resaltado de sintaxis, autocompletado inteligente, depuración integrada y control de versiones con Git. Su interfaz intuitiva y personalizable lo convierte en una excelente opción para proyectos de ciencia de datos y desarrollo web.

F. Kaggle para el conjunto de datos

Kaggle es una plataforma en línea que alberga conjuntos de datos y competencias de ciencia de datos. En este proyecto, utilizamos el conjunto de datos "Palmer Penguins" disponible en Kaggle.

Kaggle es una excelente fuente para obtener conjuntos de datos de muestra y reales para practicar y aplicar técnicas de ciencia de datos. Además, ofrece una comunidad activa de científicos de datos y recursos útiles, como kernels y notebooks.

G. GitHub como plataforma de desarrollo colaborativo

GitHub es una plataforma de desarrollo colaborativo basada en Git, un sistema de control de versiones ampliamente utilizado. En este proyecto, utilizamos GitHub para alojar nuestro código fuente, colaborar en equipo y realizar un seguimiento de los cambios y versiones del proyecto.

GitHub facilita el trabajo en equipo, permitiendo a los desarrolladores revisar, fusionar y administrar el código de manera eficiente. Además, ofrece características adicionales, como la creación de problemas, solicitudes de extracción y revisiones de código, lo que mejora el flujo de trabajo de desarrollo.

IV. PROCESO DE DESARROLLO E IMPLEMENTACIÓN

El proceso de desarrollo e implementación del proyecto se diseñó para garantizar una aplicación web interactiva de ciencia de datos eficiente y eficaz utilizando el conjunto de datos "Palmer Penguins". A través de un enfoque estructurado y colaborativo, el objetivo fue lograr una integración perfecta de las diferentes etapas, desde el análisis de datos y el desarrollo hasta la implementación y el despliegue de la aplicación.

A. Obtención de los datos

El conjunto de datos "Palmer Penguins" fue recopilado por el ecólogo Dr. Kristen Gorman y su equipo en la Isla Torgersen, cerca de la Base Palmer en la Antártida. Contiene mediciones de tres especies de pingüinos: Adelie, Gentoo y Chinstrap. Las variables incluyen medidas físicas como longitud del pico, profundidad del pico, longitud de la aleta y peso, así como información sobre la especie y la región de anidación. Este conjunto de datos valioso nos permitió explorar patrones, diferencias y características únicas entre las especies de pingüinos.

B. Limpieza y división del conjunto de datos

En el proceso de limpieza de datos, primero cargamos el conjunto de datos y realizamos comprobaciones para identificar los valores que faltaban y los tipos de datos de cada columna. E utilizó la biblioteca de Python Pandas para una manipulación eficiente de datos y operaciones de limpieza. Identificamos columnas categóricas como nombres de especies y regiones de anidación para un análisis posterior.

Después de completar todas las etapas de limpieza y preparación de datos, exportamos un nuevo conjunto de datos con datos limpios y correctamente estructurados. Este conjunto de datos limpio estuvo listo para su uso en análisis estadístico, modelado predictivo y visualizaciones, proporcionando información confiable y precisa.

C. Entrenamiento del modelo de aprendizaje automático

Para el entrenamiento del modelo, definimos variables clave como longitud del pico, profundidad del pico, longitud de la aleta y peso. Creamos una canalización para integrar las etapas de preprocesamiento, entrenamiento y evaluación de modelos, y la entrenamos con el conjunto de datos limpio.

Utilizamos un modelo de bosque de aislamiento para detectar anomalías en las mediciones, lo cual es fundamental para identificar patrones inusuales. Las predicciones de estas anomalías se utilizaron para generar alertas automatizadas.

Además, empleamos un modelo de agrupamiento de K-medias para agrupar los datos en clústeres en función de similitudes. Los resultados del análisis de clustering se visualizaron en un gráfico 3D dentro de la aplicación, permitiendo explorar los datos desde una perspectiva espacial y temporal.

El objetivo del modelo entrenado fue predecir características de los pingüinos en función de los datos de entrada. Al aprovechar las técnicas de aprendizaje automático y análisis de datos, el sistema tiene como objetivo proporcionar información precisa y oportuna sobre las características de los pingüinos, promoviendo la comprensión y la conciencia ambiental.

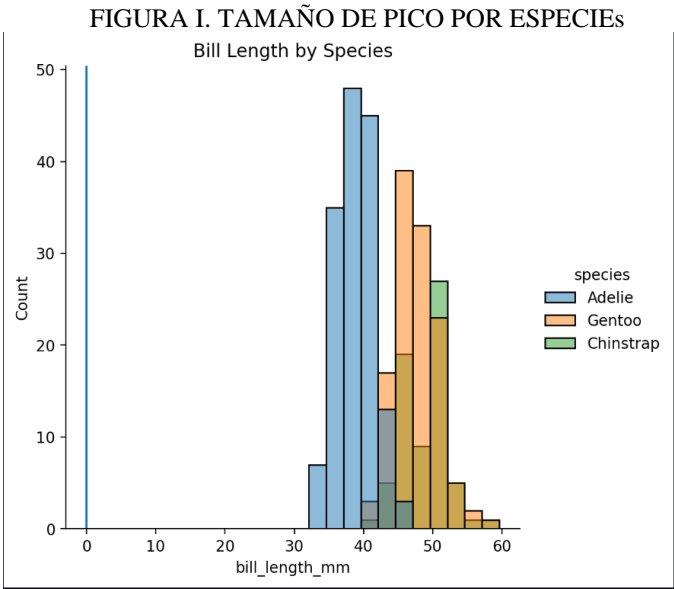
Finalmente, exportamos el modelo entrenado utilizando joblib para integrarlo en las herramientas de visualización de datos, lo que permitió una exploración completa e interactiva del conjunto de datos y facilitó una mejor toma de decisiones.

D. Visualización de datos

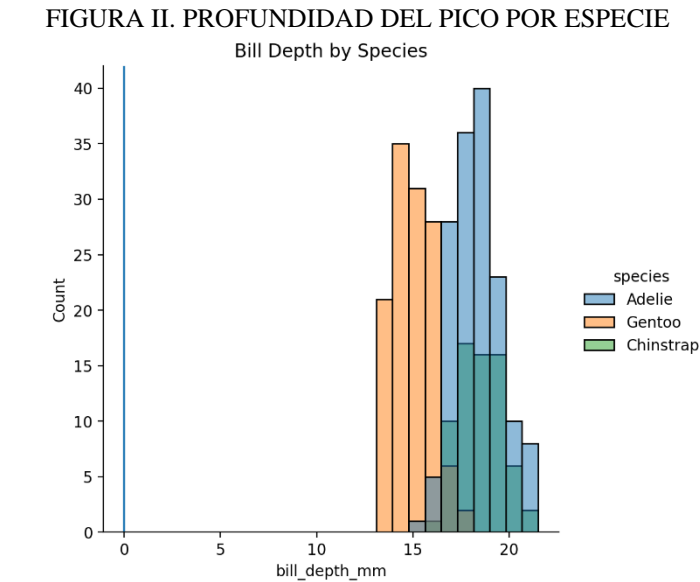
En esta sección, nos enfocamos en presentar los datos del conjunto "Palmer Penguins" de manera visualmente atractiva e interactiva. Utilizamos la biblioteca Streamlit de Python para desarrollar una aplicación web que permite explorar y visualizar los datos de forma dinámica.

Tablas interactivas Una de las características clave de nuestra aplicación son las tablas interactivas que muestran diferentes medidas de los pingüinos agrupadas por especie. Estas tablas permiten a los usuarios examinar y comparar fácilmente los valores de cada variable para las tres especies: Adelie, Gentoo y Chinstrap.

a. **Tamaño del pico por especie:** Esta tabla muestra los valores de la longitud del pico (en milímetros) para cada especie de pingüino. Los usuarios pueden observar rápidamente las diferencias en el tamaño del pico entre las especies y explorar los valores mínimos y máximos.



b. **Profundidad del pico por especie:** De manera similar, esta tabla presenta la profundidad del pico (en milímetros) agrupada por especie. Esta medida puede ser útil para identificar patrones y diferencias en la forma del pico entre las especies de pingüinos.



c. **Longitud de la aleta por especie:** La longitud de la aleta (en milímetros) también se muestra en una tabla separada, agrupada por especie. Esta variable puede proporcionar información valiosa sobre la capacidad de natación y movimiento de los pingüinos.

FIGURA III. LONGITUD DE LA ALETA POR SPECIE

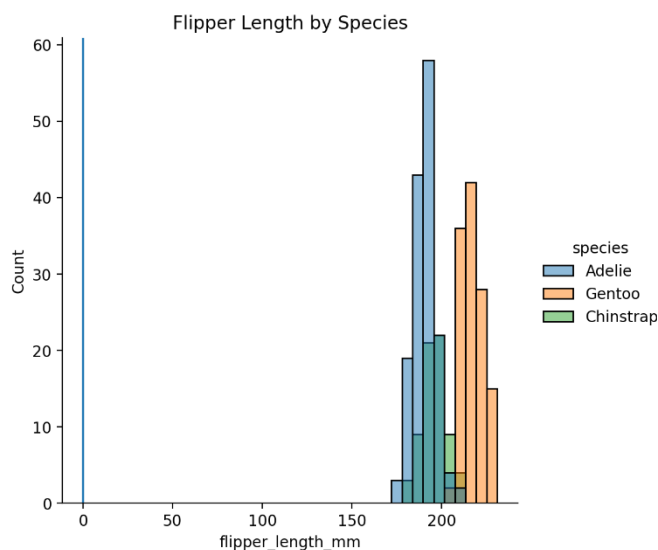
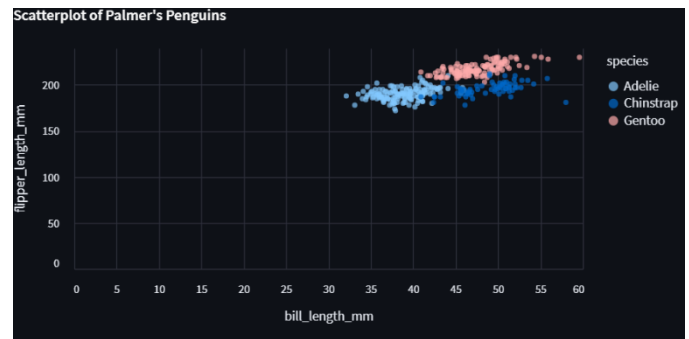
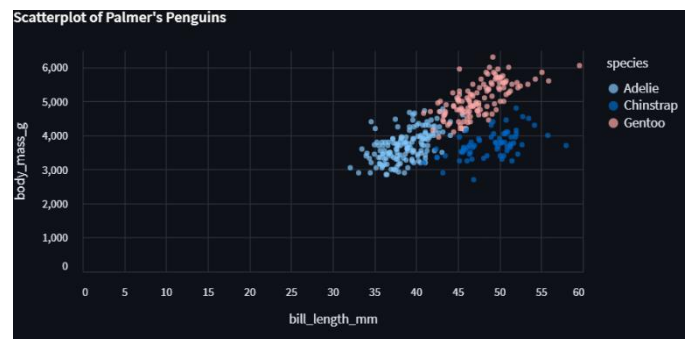


FIGURA V. GRÁFICOS DISPERSOS LONGITUD DEL PICO VS LONGITUD DE LA ALETA



c. **Peso vs. Longitud de la aleta:** Este gráfico de dispersión relaciona el peso de los pingüinos con la longitud de la aleta. Puede ser útil para estudiar la influencia del tamaño del pingüino en su capacidad de movimiento y natación.

FIGURA VI. GRÁFICOS DISPERSOS PESO VS LONGITUD DE ALETA



Todos estos gráficos de dispersión son interactivos, lo que significa que los usuarios pueden hacer zoom, desplazarse y seleccionar puntos de datos individuales para obtener más información. Además, se pueden aplicar filtros para examinar subconjuntos específicos de los datos.

Estas visualizaciones interactivas permiten a los usuarios explorar y comprender mejor el conjunto de datos "Palmer Penguins", identificar patrones y tendencias, y obtener información valiosa sobre las características y comportamientos de las diferentes especies de pingüinos.

E. Contenedores e imágenes

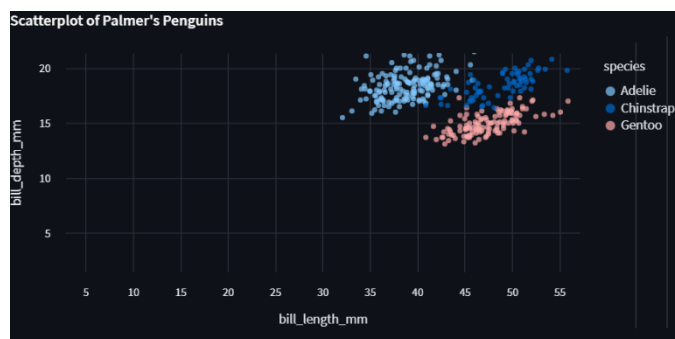
Para la creación de la imagen de la aplicación, se utilizó la plataforma Docker Desktop, que proporciona un entorno de desarrollo e implementación para aplicaciones basadas en contenedores Docker. El proceso de creación de la imagen comenzó con la elaboración de un Dockerfile, un documento que contiene instrucciones para construir la imagen. En este Dockerfile se describen los pasos necesarios para configurar el entorno de la aplicación, incluida la instalación de

Gráficos de dispersión interactivos

Además de las tablas, nuestra aplicación web incluye una variedad de gráficos de dispersión interactivos que permiten visualizar las relaciones entre diferentes variables del conjunto de datos.

a. **Longitud del pico vs. Profundidad del pico:** Este gráfico de dispersión muestra la relación entre la longitud y la profundidad del pico para cada individuo. Los usuarios pueden seleccionar las especies que desean visualizar y explorar patrones o agrupaciones interesantes.

FIGURA IV. GRÁFICOS DISPERSOS LONGITUD DEL PICO VS PROFUNDIDAD DEL PICO



b. **Longitud del pico vs. Longitud de la aleta:** De manera similar, este gráfico de dispersión compara la longitud del pico con la longitud de la aleta. Los usuarios pueden identificar posibles correlaciones entre estas variables y examinar las diferencias entre las especies.

dependencias, la definición de variables de entorno y la copia de archivos para su funcionamiento.

Una vez que se completó el Dockerfile, la imagen de la aplicación se construyó mediante el comando 'docker build'. Este proceso compiló la imagen en función de las instrucciones definidas en el Dockerfile, lo que garantiza la reproducibilidad y la coherencia del entorno de la aplicación en cualquier sistema en el que se implemente.

Al compilar la imagen, se creó un contenedor de Docker para almacenar y ejecutar la aplicación. Los contenedores Docker proporcionan un entorno aislado y ligero para la ejecución de aplicaciones, simplificando su gestión y escalabilidad. La aplicación se ejecutó dentro del contenedor mediante el comando 'docker run', lo que garantizó que todas las dependencias y configuraciones se establecieran correctamente.

Después de ejecutar la imagen en el contenedor y verificar su funcionalidad, la imagen se cargó en el repositorio de Docker Hub. Docker Hub es un servicio en la nube que permite almacenar y compartir imágenes de contenedores de Docker de forma pública o privada. Cargar la imagen en Docker Hub facilita su distribución e implementación en otros entornos, lo que permite que otros desarrolladores y equipos accedan y utilicen la aplicación de manera fácil y eficiente.

F. Implementación de aplicaciones

La aplicación web se implementó utilizando Amazon Web Services (AWS) y Amazon Lightsail durante un período de prueba de 30 días, que expirará el 23 de mayo de 2024. El proceso de implementación de una aplicación en AWS suele implicar varios pasos. Inicialmente, se crea una instancia en Amazon Lightsail, que proporciona un servidor privado virtual (VPS) preconfigurado equipado con recursos informáticos, de almacenamiento y de red. Esta instancia sirve como entorno de hospedaje para la aplicación.

Una vez configurada la instancia, se puede acceder a ella a través de Secure Shell (SSH) para la configuración y personalización de la aplicación. Esto incluye la instalación de las dependencias necesarias, como bibliotecas y marcos, la carga del código base de la aplicación y la configuración de los ajustes para alinearse con los requisitos de la aplicación. Por motivos de seguridad, se establecen reglas de firewall y se asigna una dirección IP pública para permitir el acceso externo a la aplicación.

Después de configurar el entorno, se realizan pruebas exhaustivas para garantizar que la aplicación funcione correctamente y cumpla con los estándares de rendimiento. Esta fase de prueba abarca pruebas de funcionalidad, pruebas de carga para evaluar la escalabilidad y pruebas de seguridad para identificar y abordar vulnerabilidades. Una vez que la aplicación supera estas pruebas, está lista para la implementación en producción y se configuran herramientas de supervisión para realizar un seguimiento de su rendimiento, disponibilidad e interacciones con el usuario. El mantenimiento y las actualizaciones periódicas también

forman parte del proceso continuo para garantizar una funcionalidad y una experiencia de usuario óptimas a lo largo del ciclo de vida de la aplicación en AWS. El enlace de la aplicación web es: <https://container-service-2.tfnlabkhm86og.us-east-1.cs.amazonlightsail.com>

G. Gestión de proyectos de almacenamiento y software

Para la gestión del sistema a través del control de versiones, se utilizó la plataforma GitHub. GitHub es ampliamente reconocido por su funcionalidad para controlar y rastrear cambios en el código fuente de proyectos de software. Permite a los desarrolladores colaborar de manera eficiente al proporcionar herramientas para el control de versiones, la administración de problemas, el seguimiento de tareas y la revisión de código a través de solicitudes de incorporación de cambios. Además, GitHub ofrece características como ramas para el desarrollo paralelo de características, integración continua (CI) para automatizar las pruebas y la implementación, y la capacidad de mantener un historial detallado de todos los cambios de código, lo que facilita la colaboración y el mantenimiento del sistema a lo largo del tiempo. El enlace a este proyecto se puede encontrar:

https://github.com/ariel930330/proyectofinal_cloud_computer

CONCLUSIONES

A través del desarrollo de esta aplicación web interactiva de ciencia de datos, hemos logrado crear una herramienta poderosa para explorar y analizar el conjunto de datos "Palmer Penguins". Mediante la combinación de técnicas de análisis de datos, aprendizaje automático y visualización, hemos obtenido una comprensión más profunda de las características y patrones de las diferentes especies de pingüinos.

Una de las principales conclusiones de este proyecto es la efectividad de las técnicas de aprendizaje automático para identificar anomalías y patrones en los datos. El modelo de bosque de aislamiento entrenado fue capaz de detectar mediciones inusuales o inesperadas, lo cual es fundamental para monitorear la calidad del agua y las condiciones ambientales en las áreas de anidación de los pingüinos.

Además, el análisis de clustering mediante el modelo de K-medias nos permitió agrupar los datos en clústeres basados en similitudes, revelando patrones y tendencias distintos en múltiples ubicaciones y a lo largo del tiempo. Esta capacidad de segmentar los datos de manera efectiva puede ser invaluable para comprender mejor los factores que influyen en las características y comportamientos de los pingüinos.

Las visualizaciones interactivas desarrolladas con Streamlit han sido fundamentales para comunicar nuestros hallazgos de manera clara y accesible. Las tablas y gráficos de dispersión nos han permitido explorar las relaciones entre variables clave, como la longitud del pico, la profundidad del pico, la longitud de la aleta y el peso de los pingüinos. Estas visualizaciones han revelado diferencias significativas entre las especies y han facilitado la identificación de patrones y correlaciones interesantes.

Otro aspecto importante de este proyecto ha sido la implementación y despliegue de la aplicación web en la nube utilizando tecnologías como Amazon Web Services (AWS), Amazon Lightsail y Docker. Esto ha garantizado que nuestra aplicación sea accesible desde cualquier lugar y que se ejecute de manera consistente en diferentes entornos, facilitando la colaboración y el intercambio de conocimientos.

En resumen, este proyecto ha demostrado el poder de combinar técnicas de ciencia de datos, aprendizaje automático y visualización para obtener información valiosa a partir de conjuntos de datos. Hemos logrado crear una herramienta interactiva y visualmente atractiva que puede ser utilizada por investigadores, científicos y entusiastas de los pingüinos para comprender mejor las características y patrones de estas fascinantes aves. Esperamos que este proyecto inspire futuras investigaciones y aplicaciones en el campo de la ciencia de datos y la conservación de la vida silvestre.

REFERENCIAS

- [1] A. Gorman, "Palmer Penguins dataset," Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/parulpandey/palmer-penguins>. [Accessed: 20-May-2023].
- [2] W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 2nd ed. Sebastopol, CA: O'Reilly Media, Inc., 2017.
- [3] T. Richards, Streamlit for Data Science: Create Interactive Data Apps in Python. Shelter Island, NY: Manning Publications, 2022.
- [4] A. Ratnayake, Docker for Data Science: Building Scalable and Extensible Data Infrastructure Around the Jupyter Notebook Server. Birmingham, UK: Packt Publishing Ltd, 2021.
- [5] J. VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data. Sebastopol, CA: O'Reilly Media, Inc., 2016.
- [6] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd ed. Sebastopol, CA: O'Reilly Media, Inc., 2019.
- [7] "Amazon Web Services (AWS) - Cloud Computing Services," Amazon Web Services, Inc. [Online]. Available: <https://aws.amazon.com/>. [Accessed: 20-May-2023].
- [8] "Amazon Lightsail Documentation," Amazon Web Services, Inc. [Online]. Available: <https://docs.aws.amazon.com/lightsail/latest/lightsailuserguide/>. [Accessed: 20-May-2023].
- [9] "Docker Documentation," Docker Inc. [Online]. Available: <https://docs.docker.com/>. [Accessed: 20-May-2023].
- [10] "Pandas Documentation," The Pandas Development Team. [Online]. Available: <https://pandas.pydata.org/docs/>. [Accessed: 20-May-2023].
- [11] "NumPy Documentation," The NumPy Community. [Online]. Available: <https://numpy.org/doc/>. [Accessed: 20-May-2023].
- [12] "Scikit-learn Documentation," The Scikit-learn Developers. [Online]. Available: <https://scikit-learn.org/stable/documentation.html>. [Accessed: 20-May-2023].
- [13] "Streamlit Documentation," Streamlit Inc. [Online]. Available: <https://docs.streamlit.io/>. [Accessed: 20-May-2023].