**Informe: Procesamiento de Datos Sísmicos con Detección de Anomalías y Almacenamiento en Base de Datos**

Autor: Walter Lopez Valiente

Carrera: Ing Informatica

**Introducción**

Este informe describe un proyecto académico desarrollado como parte de mi formación en programación y análisis de datos. El objetivo del proyecto fue crear un programa en Python capaz de procesar datos sísmicos almacenados en un archivo CSV, detectar anomalías en dichos datos utilizando una red neuronal simple (autoencoder) y almacenar tanto los datos originales como los resultados de la detección en una base de datos relacional (SQLite). Además, se elaboró un modelo entidad-relación (ER) para representar la estructura de la base de datos utilizada.Este proyecto me permitió aplicar conceptos clave de programación, inteligencia artificial y bases de datos, áreas fundamentales en mi desarrollo como estudiante de segundo año de universidad

**Descripción del Código**

El programa desarrollado realiza las siguientes tareas principales:

**Lectura de Datos:**

El código lee un archivo CSV llamado allhitexp.csv, que contiene datos sísmicos con nueve columnas: longitud, latitud, rate1, rate2, rate3, rate4, rate5, rate6 y ocurre. Estas columnas representan información geográfica y tasas asociadas a eventos sísmicos, así como una columna adicional (ocurre) que podría indicar la ocurrencia de un evento específico.

**Preprocesamiento de Datos**:

Para preparar los datos para el análisis, el programa selecciona automáticamente las columnas numéricas (como longitud, latitud, y las tasas rate1 a rate6). Estas características se normalizan utilizando StandardScaler de la biblioteca Scikit-learn, lo que asegura que todas las variables tengan la misma escala y no afecten negativamente el rendimiento del modelo.

**Detección de Anomalías con Autoencoder:**

Se implementa un autoencoder, una red neuronal simple, para aprender a reconstruir los datos de entrada. El autoencoder consta de tres capas: una capa de entrada con tantas neuronas como características numéricas, una capa de codificación con dos neuronas y una capa de salida que reconstruye los datos originales.

Después de entrenar el modelo, se calcula el error de reconstrucción para cada evento. Si este error supera un umbral (calculado como la media más tres veces la desviación estándar de los errores), el evento se clasifica como una anomalía.

**Almacenamiento en Base de Datos:**

Los datos originales y los resultados de la detección de anomalías se almacenan en una base de datos SQLite llamada eventos.db. La base de datos contiene dos tablas:

**•events**: Almacena los datos sísmicos originales, incluyendo las nueve columnas del archivo CSV.

**•anomalies**: Almacena los resultados de la detección, vinculados a la tabla events mediante una clave foránea (event\_id).

El código está estructurado en funciones modulares, lo que facilita su comprensión y mantenimiento. Cada función se encarga de una tarea específica, como la lectura de datos, el preprocesamiento, el entrenamiento del modelo y la inserción en la base de datos.

**Modelo Entidad-Relación (ER)**

El modelo entidad-relación (ER) representa la estructura de la base de datos utilizada en el proyecto. La base de datos consta de dos entidades (tablas) relacionadas:

**Entidad events**:

**•Atributos**:

id (clave primaria, tipo entero)

longitud (real)

latitud (real)

rate1 (real)

rate2 (real)

rate3 (real)

rate4 (real)

rate5 (real)

rate6 (real)

ocurre (entero)

Esta tabla almacena los datos sísmicos originales leídos del archivo CSV.

**Entidad anomalies:**

**•Atributos**:

id (clave primaria, tipo entero)

event\_id (clave foránea que referencia a id en events, tipo entero)

is\_anomaly (entero, indica si el evento es una anomalía)

error (real, error de reconstrucción del autoencoder)

Esta tabla almacena los resultados de la detección de anomalías para cada evento.

**Relación:**

La relación entre events y anomalies es de uno a muchos (1:N). Un evento en la tabla events puede estar asociado a múltiples entradas en anomalies (aunque en este proyecto, cada evento tiene solo una entrada de anomalía), mientras que cada entrada en anomalies pertenece a un solo evento en events. Esta relación se establece mediante la clave foránea event\_id en la tabla anomalies.

El diagrama ER se encuentra en el apéndice de este informe para su revisión.

**Herramientas y Tecnologías Utilizadas**

El proyecto integró varias herramientas y tecnologías relevantes en el campo de la ciencia de datos y la inteligencia artificial:

**Python**: Lenguaje de programación utilizado para desarrollar el código.

**Pandas**: Biblioteca para la manipulación y lectura de datos en formato CSV.

**Scikit-learn**: Utilizada para la normalización de datos mediante StandardScaler.

**TensorFlow/Keras**: Frameworks para la construcción y entrenamiento del autoencoder.

**SQLite**: Sistema de gestión de bases de datos ligero utilizado para almacenar los datos y resultados.

Estas herramientas fueron seleccionadas por su simplicidad y eficacia, lo que las hace ideales para un proyecto académico de este nivel.

**Reflexión y Aprendizaje**

Este proyecto me permitió aplicar y consolidar varios conceptos clave:

**Programación en Python**: Aprendí a estructurar un programa de manera modular, dividiendo el código en funciones específicas para mejorar la legibilidad y mantenibilidad.

**Inteligencia Artificial**: Implementé un autoencoder, una red neuronal simple, para realizar una tarea de detección de anomalías sin necesidad de etiquetas previas, lo que me introdujo al aprendizaje no supervisado.

**Bases de Datos**: Diseñé y utilicé una base de datos relacional, comprendiendo cómo establecer relaciones entre tablas mediante claves primarias y foráneas.

**Análisis de Datos**: Apliqué técnicas de preprocesamiento, como la normalización, para preparar los datos para el modelo.

Este ejercicio fue fundamental para mi desarrollo académico, ya que combinó teoría y práctica en áreas esenciales como la programación y el análisis de datos, preparándome para desafíos más complejos en el futuro.

**Conclusión**

El proyecto cumplió con éxito los objetivos planteados: procesar datos sísmicos, detectar anomalías utilizando un autoencoder y almacenar la información en una base de datos relacional. El modelo entidad-relación (ER) proporcionó una representación clara de la estructura de la base de datos, facilitando la comprensión de las relaciones entre los datos.Este tipo de proyectos es crucial para mi formación, ya que me permite aplicar conceptos teóricos en situaciones prácticas, desarrollando habilidades que serán valiosas tanto en mi trayectoria académica como profesional

**Apéndice: Diagrama del Modelo Entidad-Relación (ER)**

|  |
| --- |
| **Events** |
| id (PK) |
| longitud |
| latitud |
| rate 1  1  N |
| rate 2 |
| rate 3 |
| rate 4 |
| rate 5 |
| rate 6 |
| ocurre |

|  |
| --- |
| **Anomalies** |
| id (PK) |
| event\_id (FK) |
| is\_anomaly |
| error |

•(PK): Clave primaria

•(FK): Clave foránea

Este informe, junto con el diagrama ER, proporciona una visión completa del proyecto y demuestra la aplicación práctica de los conceptos aprendidos en el curso. Si el profesor requiere más detalles o ajustes, estaré encantado de realizarlos.