

Manual de usuario para la aplicación de predicción de defectos

Introducción

Esta aplicación desarrollada en **Streamlit** utiliza aprendizaje automático para predecir si un lote de productos será defectuoso o no, calculando en la cantidad de productos en el lote y el tiempo de entrega. También registra cada predicción realizada en un archivo CSV para su posterior análisis.

A continuación, se detalla cómo utilizar cada funcionalidad.

Requisitos previos

1. **Python 3.8 o superior** : Asegúrese de tener Python instalado en su sistema.
2. **Dependencias** : Instala los paquetes necesarios ejecutando:

```
pip install streamlit pandas matplotlib scikit-learn openpyxl
```

3. **Archivos necesarios** :
 - **Archivo de datos** : Un archivo en formato Excel (`.xlsx`) que contiene las columnas:
 - **Productos-Lote**: Cantidad de productos en el lote.
 - **Tiempo-Entrega**: Tiempo de entrega en minutos.
 - **Defectuoso**: Indicador binario (1 para defectuoso, 0 para no defectuoso).
 - **Código fuente** : Los scripts de la aplicación y las funciones auxiliares.

Iniciar la aplicación


Ejecuta el siguiente comando en tu terminal para iniciar la aplicación:

```
streamlit run app.py
```

La aplicación se abrirá automáticamente en tu navegador.


Funcionalidades

1. Cargar datos 📁

- Navega a la sección "  Carga Datos" desde la barra lateral.
- Sube un archivo Excel con los datos requeridos.
- Una vez cargados, recibirás un mensaje de confirmación.



Nota : Este paso es obligatorio antes de explorar datos o entrenar el modelo.

2. Explorar datos

- Selecciona la sección "  Explorar Datos".
- La aplicación te lo muestra:
 - Información general del conjunto de datos.
 - Las primeras filas de los datos cargados.
 - Cantidad de valores faltantes por columna.
 - Estadísticas descriptivas de las variables numéricas.


Si no se han cargado los datos, aparecerá un mensaje de advertencia.

3. Entrenar Modelo

- Ve a la sección "  Entrenar Modelo".
- Configure los parámetros del modelo desde la barra lateral:
 - **Tasa de aprendizaje** : Controla la magnitud de los ajustes en cada iteración.
 - **Número máximo de iteraciones** : Defina cuántas veces el modelo entrenará sobre los datos.
- Haz clic en "  **Entrenar Modelo**" para iniciar el entrenamiento.
- Resultados del entrenamiento:
 - **Precisión durante el entrenamiento** : Gráfico que muestra cómo evoluciona la precisión en cada iteración.
 - **Evolución de la pérdida** : Gráfico que ilustra el comportamiento de la pérdida (log-loss).
 - **Distribución de predicciones** : Histograma de las predicciones realizadas en el conjunto de entrenamiento.

Nota : Cargar datos es un paso previo necesario.

4. Predicción

- Dirígete a la sección "  predicción".
- Introduzca los siguientes datos:
 - **Cantidad de productos en el lote** .
 - **Tiempo de entrega (en minutos)** .

- Haz clic en el botón " 🧙 Predecir" .
- Resultados:
 - La aplicación mostrará si el lote es "**defectuoso**" o "**no defectuoso**" .
 - Automáticamente se registrará la predicción en un archivo llamado `registro_predicciones.csv`.

Registro de cotizaciones :

- Cada predicción se guarda en un archivo CSV con los siguientes campos:
 - Fecha y hora de la predicción.
 - Cantidad de productos en el lote.
 - Tiempo de entrega.
 - Resultado de la predicción.

Estructura del proyecto

El proyecto está organizado en varios módulos para facilitar su mantenimiento y expansión:

- **app.py** : Archivo principal que ejecuta la aplicación Streamlit.
- **utils/data_loader.py** : Función para cargar datos desde un archivo Excel.
- **utils/exploration.py** : Función para explorar y visualizar los datos.
- **utils/preprocessor.py** : Función para preprocesar los datos y dividirlos en conjuntos de entrenamiento y prueba.
- **utils/model.py** : Funciones para entrenar el modelo y realizar predicciones.

Posibles errores y soluciones

1. **Error al cargar datos :**
 - Verifique que el archivo sea en formato `.xlsx` que contenga las columnas requeridas.
2. **Predicción no disponible :**
 - Asegúrese de haber entrenado el modelo antes de intentar realizar predicciones.
3. **Problemas al registrador predicciones :**
 - Revise que tenga permisos de escritura en el directorio donde se ejecuta la aplicación.

Personalización

Para personalizar o extender la aplicación:

- Agrega nuevas columnas al conjunto de datos y actualiza las funciones de preprocesamiento.
- Cambia el modelo de regresión logística por otro algoritmo de clasificación (como árboles de decisión o redes neuronales).
- Agregue nuevas funcionalidades en el archivo `app.py`.