

En base a:

Rubrica del Bootcamp de Machine Learning

Proyecto Final

Rúbrica de evaluación

Definición del problema y objetivos (20%)

- Claridad en la descripción del problema a resolver.
- Justificación de la relevancia del problema y su impacto potencial.
- Identificación de los stakeholders clave para la solución del problema
- Establecimiento de objetivos claros y alcanzables.
- Análisis de las acciones que se podrían tomar antes de comenzar a crear un modelo de machine learning.
- Documentación sobre los datos (obtención y principales características sobre los mismos).
- Documentación sobre el tipo y la metodología de despliegue propuesta.

Análisis y preprocesamiento de datos (25%)

- Realización de un análisis exploratorio adecuado de los datos.
- Limpieza y manejo adecuado de datos faltantes, duplicados o inconsistentes.
- Aplicación de técnicas de feature engineering y feature selection.

Modelado y evaluación (25%)

- Selección apropiada de técnicas de modelado según las restricciones y requisitos técnicos del problema.
- Creación de un baseline para comparar modelos nuevos.
- El modelo fue evaluado con al menos dos métricas adecuadas para el tipo de problema (por ejemplo, accuracy, F1, ROC AUC, etc.), y se justifica la elección de dichas métricas.
- Documentación sobre el análisis de errores que se llevó a cabo para el modelo creado.

Implementación de un pipeline y despliegue de modelos (20%):

- Implementación exitosa de un pipeline de entrenamiento que comience con la extracción de los datos y culmine con un modelo ya entrenado.
- Uso de un sistema de registro de experimentos y/o modelos.
- Documentación clara y detallada del pipeline de entrenamiento.

Creatividad y comunicación (10%)

- Incorporación de ideas originales o enfoques novedosos en el proyecto.
- Implementación de técnicas o métodos adicionales que no se cubrieron explícitamente en el curso.
- Documentación clara y detallada de los procesos del proyecto.

Extras (+20%)

- Uso de control de versiones para código y datos
- Implementación de automatización de pipelines de entrenamiento o inferencia.
- Uso de herramientas para el monitoreo de modelos en producción.
- Propuesta de cómo recolectar el feedback para mejorar continuamente el modelo.

Fecha de entrega: 25 de Agosto de 2025

Mi Proyecto: Un Modelo de Siembra Inteligente para Productores

URL pública de la app:

<https://huggingface.co/spaces/DoraCastillo/planificador-siembra>

Video de presentacion:

<https://youtu.be/Kge0P5N3 SQ>

1. El Problema

Un productor necesita una forma precisa de planificar la siembra para asegurar la producción necesaria para su familia y ganado. Las estimaciones manuales son imprecisas y pueden llevar a un déficit o, peor aún, a una pérdida de cosecha. Mi proyecto busca resolver este problema, transformando la incertidumbre en una ***recomendación informada***.

2. La Solución: Un Sistema Inteligente

Mi solución es un Producto Mínimo Viable (**MVP**): una calculadora de siembra impulsada por inteligencia artificial que ofrece una recomendación clara y accionable. La clave del MVP es que la calculadora utiliza los rendimientos predichos por mi modelo de machine learning en lugar de valores fijos o estimados. Esto hace que la herramienta sea más precisa y confiable para el productor.

El flujo de trabajo completo es el siguiente:

1. Entrada de Datos Simples: El productor solo necesita proporcionar unos pocos datos que ya conoce, como las condiciones climáticas o los requerimientos de consumo.
2. Motor de Cálculo Inteligente: Internamente, el sistema utiliza el rendimiento predicho por mi modelo de IA para calcular la demanda total y las hectáreas de siembra necesarias.
3. Salida Clara y Útil: El resultado final es una tabla que muestra la cantidad de hectáreas recomendada para sembrar de cada cultivo. Esto le da al productor una cifra concreta para su planificación.

3. Dataset y Fuentes de Datos

Para esta demo, decidí construir un dataset sintético y reproducible que combina los datos esenciales del sector agrícola. Mi objetivo era garantizar que mi presentación fuera fluida y sin errores, por lo que creé mis propias fuentes de datos basadas en una investigación exhaustiva.

- **Datos Climáticos:** Simulé variables clave como la precipitación anual y la temperatura media, basándome en los patrones reales de la región. Mi investigación se centró en fuentes como el SMN (Servicio Meteorológico Nacional) y proyectos de datos globales como CHIRPS.
- **Datos Agrícolas:** Generé los rendimientos de los cultivos de Maíz, Choclo y Zapallo a partir de funciones que replican las relaciones reales entre el clima y la producción.
- **Fuentes:** Para asegurar la validez de mis datos simulados, me basé en información de bases de datos como ScienceDirect y las estimaciones agrícolas publicadas por el Ministerio de Agricultura, Ganadería y Pesca de Argentina.

4. Características Clave (Features)

Para asegurar que mi modelo fuera robusto, creé un conjunto de características diseñadas para capturar los factores más importantes que afectan el crecimiento de los cultivos.

- **Precipitación Acumulada:** Calculé el total de lluvia anual, un factor crítico en mi modelo.
- **Temperatura Media Anual:** Obtuve la temperatura promedio, que es crucial para el desarrollo de los cultivos.
- **Estado ENSO:** Incluí un índice para el fenómeno El Niño/La Niña, ya que su impacto en el clima de la región es significativo.
- **NDVI Acumulado:** Utilicé esta característica para simular la "salud de la vegetación" a lo largo del año.
- **Grados-Día de Crecimiento (GDD):** Implementé el GDD como una medida de la acumulación de calor, vital para el ciclo de vida de las plantas.
- **Precios y Costos:** Agregué variables económicas para el componente de optimización de mi proyecto.

5. Modelos y Visualización

El corazón de mi proyecto se compone de dos modelos principales y una visualización final que demuestran la funcionalidad de mi herramienta.

- **Modelos de Predicción:** Utilicé un modelo de Random Forest Regressor como mi base. Elegí este modelo por su solidez y capacidad para manejar relaciones no lineales.
- **Modelo de Optimización:** Este es el motor de mi demo. Construí un modelo de optimización que utiliza los rendimientos que predije para calcular la asignación de hectáreas de cada cultivo que me permitirá maximizar la ganancia, mientras cumpla con las restricciones de demanda y de superficie total.
- **Visualización Final:** El resultado de la optimización es un gráfico de barras que muestra mi recomendación de siembra y una cifra clara de la ganancia total proyectada. Esta visualización es la conclusión tangible de mi proyecto.

6. Caso de Uso :

Mi proyecto se centra en la historia de un "día en la vida" de un pequeño productor. Mi stakeholder es un productor en una zona rural de Santiago del Estero que depende del autoabastecimiento y el canje con sus vecinos.

Mi demo comienza con la pregunta: "¿Qué sembrar este año?" Para el productor, el objetivo principal no es solo la ganancia, sino la seguridad. Por lo tanto, utilicé un escenario de sequía, y mi modelo no solo predice los rendimientos, sino que también prioriza la siembra de cultivos más resistentes, como el Zapallo, para mitigar el riesgo de pérdida de cosecha. El modelo ayuda al productor a tomar la mejor decisión, considerando tanto la subsistencia como la rentabilidad.

7. Despliegue de la Aplicación

Para que el proyecto fuera interactivo y accesible, desplegué la aplicación como un Space en Hugging Face, lo que permite que cualquier persona pueda probar la calculadora de siembra sin instalar nada localmente.

Pasos del despliegue:

1. Creación del Space
 - a. Space creado bajo el usuario DoraCastillo con el nombre planificador-siembra.
 - b. Se seleccionó Gradio como framework para la interfaz de usuario.
2. Gestión de dependencias

Todas las librerías necesarias para ejecutar el modelo y la interfaz se listaron en un archivo requirements.txt :

scikit-learn, pandas, numpy, gradio, Hugging Face instala automáticamente estas dependencias al detectar el archivo.

3. Subida de archivos

- a. Utilicé la librería huggingface_hub para subir de manera segura app.py y requirements.txt al Space.
- b. El token de acceso se mantuvo seguro mediante variables de entorno en Colab.

4. Reinicio y ejecución del Space

- a. Después de subir los archivos, reinicié el Space desde la sección Settings → Restart Space.
- b. El log confirmaba que la aplicación se inició correctamente:
- c. ===== Inicio de la aplicación el 2025-09-12 18:49:08 =====

5. Acceso a la aplicación

- a. URL pública de la app: <https://259c0217b48bf3f679.gradio.live>
- b. Cualquier usuario puede interactuar con la calculadora, ingresar sus datos de siembra y obtener recomendaciones inmediatas.

Beneficios del despliegue

- Permite **demostraciones en tiempo real** y persistencia.
- Facilita que el productor final pueda **probar la herramienta sin instalaciones ni conocimientos técnicos**.
- Hace que el proyecto sea **reproducible y escalable**, sentando la base para futuras mejoras como la conexión automática a APIs de datos climáticos y satelitales.

8. Cómo Evolucionaría mi MVP

El siguiente paso lógico para el proyecto sería automatizar la entrada de datos. El usuario no tendría que ingresar manualmente los datos técnicos como el GDD y el NDVI. En una versión más avanzada de la aplicación, el flujo de trabajo sería el siguiente:

1. **Conexión con APIs:** En lugar de que el usuario mueva un control deslizante, mi sistema se conectaría a una API (Interfaz de Programación de Aplicaciones) de servicios como la NASA o el Servicio Meteorológico Nacional.

2. **Extracción de Datos Automática:** El sistema consultaría estas APIs para obtener los datos más recientes de temperatura y de imágenes satelitales para la ubicación del productor.
3. **Procesamiento y Predicción:** La aplicación procesaría estos datos automáticamente, los pasaría a través de mi modelo de IA y generaría la recomendación de siembra.

De esta manera, el usuario final solo vería una interfaz limpia, donde simplemente ingresaría su ubicación y obtendría la recomendación de siembra, sin la necesidad de entender ni ingresar manualmente datos técnicos. Esto transforma mi MVP en un producto completamente funcional y fácil de usar.