Resuelve ML:

- identifica el tipo de suelo

- identifica tipo de semilla

- simula diferentes escenarios para realizar una predicción o estimación

- predicción de rendimiento de cultivo

Resuelve AG:

- valores óptimos de productos a utilizar

- ruta optima del riego para cubrir el área

- optimización del uso del agua de riego

- optimiza la mejor combinación

- ubicar de forma óptima los sensores en campos

- optimización de calendario de siembra y cosecha

Preguntas:

* ¿Qué es ML?
* ¿Hay diferentes tipos?
* ¿Cómo se entrena?
* ¿Cómo puedo realizar una simulación de las semillas?

Uso de ML con AG:

* El ML puede ser utilizado como sustituto de la función fitness dentro de los AG.

**Bases de datos:**

* <https://datos.magyp.gob.ar/dataset>
* <https://www.magyp.gob.ar/datosabiertos/>
* <https://datos.gob.ar/dataset?groups=agri>

**Supuestos:**

* No se toman en cuenta las plagas.

**Introducción:**

La planificación agrícola en Argentina enfrenta actualmente una creciente complejidad, impulsada por la necesidad de tomar decisiones estratégicas sobre qué cultivos implantar, cómo organizarlos espacialmente dentro de las parcelas y cómo rotarlos a lo largo del tiempo. Estos procesos no pueden abordarse de manera aislada, ya que dependen de múltiples factores interrelacionados, como las propiedades del suelo, las condiciones climáticas, la disponibilidad hídrica y las interacciones entre cultivos sucesivos. A pesar de ello, en muchos casos las decisiones productivas aún se basan en la experiencia del productor o en recomendaciones estandarizadas, sin contemplar adecuadamente las particularidades de cada unidad productiva.

Este enfoque empírico limita el aprovechamiento del potencial productivo y puede derivar en una gestión ineficiente de los recursos naturales, afectando la sostenibilidad a largo plazo de los sistemas agrícolas. Frente a esta situación, se vuelve necesario avanzar hacia estrategias de planificación basadas en datos, capaces de procesar y analizar simultáneamente una gran cantidad de variables para orientar la toma de decisiones de forma más precisa, adaptable y contextualizada.

En este escenario, los algoritmos genéticos han demostrado ser eficaces para resolver problemas de optimización complejos y con múltiples variables, como los que se presentan en la planificación agrícola. Estos algoritmos permiten explorar grandes espacios de búsqueda en busca de combinaciones óptimas de cultivos, localización y tiempos.

Paralelamente, los avances recientes en machine learning han permitido construir modelos predictivos precisos basados en datos históricos, capaces de estimar el rendimiento, la eficiencia en el uso de recursos o el riesgo agronómico de distintas decisiones. La integración de estas dos herramientas —los algoritmos genéticos como método de optimización y el machine learning como motor de predicción— ofrece una oportunidad innovadora para generar soluciones precisas, dinámicas y adaptadas a las condiciones específicas de cada unidad productiva.

Cabe señalar que, en el presente trabajo, se ha decidido acotar deliberadamente el alcance del modelo propuesto. En particular, se omiten variables relacionadas con la incidencia de enfermedades, la presencia de plagas y los niveles de contaminación ambiental o química. Esta decisión metodológica responde a la necesidad de abordar inicialmente la problemática desde una perspectiva simplificada, que permita focalizar el análisis en los aspectos estructurales de la planificación agrícola, tales como la selección de cultivos, la distribución espacial y temporal, y la consideración de factores geográficos y climáticos.

**Problemática:**

En el contexto actual de la agricultura argentina, la planificación eficiente de la producción se ha vuelto una tarea cada vez más compleja. Esta planificación no solo abarca la selección de cultivos, sino también su disposición espacial dentro de las parcelas y su rotación temporal a lo largo de múltiples temporadas. Las decisiones en torno a estos aspectos deben considerar simultáneamente múltiples factores: las características del suelo, los cambios de clima, la disponibilidad de agua, las interacciones entre cultivos sucesivos y las condiciones económicas del entorno productivo.

Pese a esta complejidad, en muchos casos las decisiones agronómicas aún se toman basadas en la experiencia previa del productor o en recomendaciones técnicas generales que no logran captar las particularidades de cada parcela ni adaptarse dinámicamente a condiciones cambiantes. Esto puede derivar en una subutilización del potencial productivo, un manejo ineficiente de los recursos naturales (especialmente agua y nutrientes del suelo) y una pérdida de sustentabilidad del sistema agropecuario a largo plazo.

Me puedo explayar y va el contexto de cómo llego al problema. Lo demás en introducción.

**Problema:**

En la actividad agrícola se toman muchas decisiones (¿qué cultivos sembrar? ¿dónde ubicar cada cultivo? ¿cuándo realizar la siembra y cosecha?), estas decisiones deben considerar múltiples factores como el tipo de suelo, el clima, la rotación de cultivos y la disponibilidad de agua. La falta de una estrategia óptima puede afectar negativamente el rendimiento, la sostenibilidad del suelo y la eficiencia en el uso de recursos. Estas variables están interrelacionadas entre sí, debido a esto y a la gran cantidad de combinaciones posibles, este problema constituye un sistema altamente complejo, por lo que se plantea abordarlo utilizando algoritmos genéticos apoyados en modelos de machine learning para encontrar soluciones óptimas o cercanas al óptimo.

Mas resumido, en una pregunta directa.

¿Cómo se puede mejorar la toma de decisiones en la agricultura?

¿Se puede desarrollar algún método que permita tomar la mejor decisión frente a una gran cantidad de variables?

¿Cómo se pueden analizar las variables climáticas, hídricas y de los recursos de los suelos, en conjunto con la siembra sucesiva y/o paralela de diferentes semillas, para obtener un resultado óptimo? Con óptimo nos referimos a que desgaste lo menos posible el suelo y genere la mayor cantidad de ganancias.

Elegidas:

* ¿Cómo aplicar modelos de machine learning combinados con algoritmos genéticos para optimizar la planificación agrícola, considerando múltiples variables interdependientes como clima, suelo y rotación de cultivos?
* ¿De qué manera pueden integrarse algoritmos genéticos y técnicas de machine learning para asistir en la toma de decisiones agrícolas complejas, como la selección, disposición y rotación de cultivos?
* ¿Es posible desarrollar una herramienta computacional basada en algoritmos genéticos y machine learning que permita optimizar la planificación agrícola en función de variables edafoclimáticas y productivas?
* ¿Cómo pueden los algoritmos genéticos, apoyados en modelos de machine learning, contribuir a mejorar la eficiencia, sostenibilidad y productividad en la toma de decisiones sobre planificación agrícola?

**Objetivos:**

General:

Desarrollar un modelo de optimización para la planificación espacial y temporal de cultivos en parcelas agrícolas, utilizando algoritmos genéticos y técnicas de machine learning.

Los algoritmos genéticos serán utilizados para explorar combinaciones óptimas de cultivos, ubicaciones, tiempo de siembre y cosecha, evaluando cada una con una función fitness que puede, por ejemplo, maximizar el rendimiento esperado y mejorar la rotación, entre otras cosas.

Se aplicarán algoritmos de machine learning para predecir el rendimiento esperado dado un conjunto de condiciones iniciales basándose en los datos obtenidos de experiencias previas. Este modelo se integraría como parte de la función fitness del algoritmo genético.

Específicos:

* Obtener y estructurar datos agronómicos relevantes, tales como tipo de cultivo, características del suelo, datos climáticos, historial de producción y rotación de cultivos, para entrenar el modelo de machine learning.
* Diseñar un modelo predictivo basado en machine learning que estime el rendimiento esperado de los cultivos bajo distintas combinaciones de condiciones climáticas y geográficas, de disposiciones en parcelas de cultivos y selecciones de semillas.
* Desarrollar un algoritmo genético que utilice como función fitness los resultados del modelo predictivo, a fin de encontrar combinaciones óptimas de distribución y rotación de cultivos.
* Desarrollar una aplicación de selección del terreno que indique los parámetros de dónde se realizará el cultivo (localidad, metros cuadrados y disposición del terreno).
* Testear el modelo de optimización propuesto mediante simulaciones, evaluando su rendimiento, eficiencia y sostenibilidad agrícola.
* Generar una herramienta de apoyo a la toma de decisiones que pueda ser utilizada por productores o técnicos agrícolas para planificar de forma óptima el uso de sus parcelas, por ejemplo: una aplicación que muestre gráficos y comparativas.

**Soluciones:**

* Cálculo de maximización de ganancias frente al sembrado de diferentes semillas.
* Cálculo de minimización de costos frente al sembrado de diferentes semillas.
* Optimización del uso del territorio.
* Optimización de rutas de riego dentro de un territorio.
* Simulación puede incluir:
  + Datos climáticos
  + Datos de la tierra

**Videos:**

* <https://www.youtube.com/watch?v=lwxgWOkgdhM&pp=ygUabWkgcHJvcGlvIG1hY2hpbmUgbGVhcm5pbmc%3D>
* <https://www.youtube.com/watch?v=OzHNEoYAKyI&pp=ygUabWkgcHJvcGlvIG1hY2hpbmUgbGVhcm5pbmfSBwkJiwkBhyohjO8%3D>
* <https://www.youtube.com/watch?v=OzHNEoYAKyI&pp=ygUabWkgcHJvcGlvIG1hY2hpbmUgbGVhcm5pbmfSBwkJiwkBhyohjO8%3D>
* <https://www.youtube.com/watch?v=iX_on3VxZzk&pp=ygUabWkgcHJvcGlvIG1hY2hpbmUgbGVhcm5pbmfSBwkJiwkBhyohjO8%3D>
* <https://www.youtube.com/watch?v=i_LwzRVP7bg&t=525s&pp=ygUabWkgcHJvcGlvIG1hY2hpbmUgbGVhcm5pbmc%3D>

**Bibliografía:**

* <https://medium.com/pytorch/ai-for-ag-production-machine-learning-for-agriculture-e8cfdb9849a1>
* <https://agrio.app/Agriculture-API/>

# Glosario:

**Funciones sigmoides:** es una función matemática que transforma cualquier valor de entrada en un rango entre 0 y 1, produciendo una gráfica con forma de "S" característica. Se usa comúnmente en redes neuronales artificiales como función de activación, para introducir no linealidad y convertir los valores de salida en probabilidades, así como en algoritmos de machine learning para tareas como la regresión logística y la clasificación binaria.

**Tangente hiperbólica:** función matemática utilizada como función de activación en redes neuronales, incluyendo las LSTM. Convierte cualquier valor real en un número entre -1 y 1, lo que permite normalizar la información y modelar relaciones no lineales. Su principal ventaja es que, a diferencia de la función sigmoide que solo devuelve valores positivos, la tangente hiperbólica produce salidas centradas en cero, lo que facilita la propagación de gradientes y contribuye a que las celdas de memoria de la LSTM puedan representar incrementos o decrementos en la información de manera más equilibrada.

# Marco teórico:

Los algoritmos genéticos serán utilizados para explorar combinaciones óptimas de cultivos, ubicaciones, tiempo de siembre y cosecha, evaluando cada una con una función fitness que puede, por ejemplo, maximizar el rendimiento esperado y mejorar la rotación, entre otras cosas.

Además, se aplicarán algoritmos de machine learning para predecir el rendimiento esperado dado un conjunto de condiciones iniciales basándose en los datos obtenidos de experiencias previas. Este modelo se integraría como parte de la función fitness del algoritmo genético. En nuestro caso, los algoritmos de Machine Learning utilizados son redes neuronales Long Short-Term Memory (LSTM) y Gradient Boosting Regression (GBM).

## AG

Los **algoritmos genéticos (AG)** son una técnica de optimización y búsqueda inspirada en los principios de la **evolución natural y la genética**. Introducidos por John Holland en la década de 1970, los AG forman parte del paradigma de la **computación evolutiva**, que busca resolver problemas complejos explorando iterativamente un espacio de soluciones mediante mecanismos inspirados en la selección natural, la recombinación y la mutación.

En esencia, un algoritmo genético trabaja sobre una **población de individuos**, donde cada individuo representa una **solución candidata** al problema a resolver. Esta representación generalmente se codifica mediante estructuras discretas, como cadenas binarias, enteros o vectores de parámetros, denominadas **cromosomas**. La calidad de cada individuo se evalúa mediante una **función de aptitud** o *fitness*, que cuantifica qué tan buena es la solución con respecto al objetivo planteado.

El ciclo básico de un algoritmo genético consiste en los siguientes pasos:

1. **Inicialización:** se genera una población inicial de manera aleatoria o basada en heurísticas, asegurando diversidad suficiente para explorar el espacio de soluciones.
2. **Evaluación:** cada individuo se evalúa mediante la función de aptitud, asignándole un valor que refleja su desempeño relativo.
3. **Selección:** se eligen los individuos que participarán en la generación siguiente. Los métodos más comunes incluyen la **ruleta** y el **torneo**, todos orientados a favorecer la supervivencia de los individuos más aptos mientras se mantiene diversidad genética.
4. **Reproducción (crossover o recombinación):** pares de individuos seleccionados combinan sus cromosomas para generar descendencia. Esta operación permite **intercambiar información genética** entre soluciones y explorar nuevas regiones del espacio de búsqueda.
5. **Mutación:** se aplican cambios aleatorios a algunos genes de la descendencia, introduciendo **variabilidad y exploración** adicional que ayuda a evitar óptimos locales.
6. **Reemplazo:** la nueva generación sustituye total o parcialmente a la anterior, y el ciclo se repite hasta que se cumple un criterio de convergencia, como un número máximo de generaciones o una solución que alcanza un valor de aptitud satisfactorio.

Una característica clave de los AG es su **capacidad de explorar espacios de soluciones altamente complejos y no lineales** sin requerir derivadas ni información explícita de la función objetivo. Esto los hace especialmente útiles en problemas donde las relaciones entre variables son inciertas, discontinuas o multidimensionales, como la optimización de rutas, el diseño de sistemas complejos, la selección de características en aprendizaje automático o la planificación agrícola.

Además, los algoritmos genéticos permiten incorporar estrategias como **elitismo**, donde los individuos más aptos se preservan de generación en generación, aumentando la probabilidad de conservar soluciones de alta calidad. Esta flexibilidad, combinada con su naturaleza estocástica, convierte a los AG en una herramienta poderosa para la resolución de problemas de optimización global en contextos donde métodos tradicionales basados en gradientes o búsqueda exhaustiva serían inviables.

## LSTM

Las redes neuronales recurrentes (RNN, por sus siglas en inglés) constituyen una clase de modelos diseñados específicamente para trabajar con datos secuenciales, como series temporales, lenguaje natural o señales biológicas. Sin embargo, las RNN tradicionales presentan limitaciones importantes al intentar capturar dependencias de largo plazo debido al problema del desvanecimiento o explosión del gradiente (Gradient Boosting) durante el entrenamiento. Para superar estas dificultades, Hochreiter y Schmidhuber (1997) propusieron la arquitectura Long Short-Term Memory (LSTM), que ha demostrado ser especialmente efectiva en el modelado de secuencias complejas.

Las redes LSTM incorporan una estructura de celdas de memoria que permiten almacenar y transmitir información relevante a lo largo de la secuencia, regulando explícitamente qué información se conserva y cuál se descarta. Esto se logra mediante tres compuertas principales:

1. **Compuerta de entrada (input gate):** controla qué información nueva debe incorporarse a la celda de memoria.
2. **Compuerta de olvido (forget gate):** determina qué fracción de la información previamente almacenada se debe descartar.
3. **Compuerta de salida (output gate):** regula qué información de la celda se utilizará para generar la salida en el paso actual.

Matemáticamente, estas compuertas se implementan como funciones sigmoides y tangentes hiperbólicas que actúan sobre los vectores de entrada y el estado oculto, permitiendo una modulación adaptativa del flujo de información.

La principal ventaja de las LSTM es su capacidad de capturar dependencias a largo plazo en series temporales, sin que el gradiente se desvanezca de forma crítica. Esto las convierte en modelos muy utilizados en tareas como:

* Predicción de series temporales (ej. variables climáticas, financieras, agrícolas).
* Traducción automática y procesamiento de lenguaje natural.
* Reconocimiento de voz y señales biomédicas.

En el contexto de predicción climática y agrícola, las LSTM resultan particularmente valiosas, ya que permiten modelar patrones estacionales y dependencias interanuales, incorporando tanto la variabilidad de corto plazo como las tendencias de largo plazo de las variables ambientales.

# GBR

El Gradient Boosting Regression (GBR) es un algoritmo de aprendizaje supervisado basado en la técnica de *boosting*, introducida por Friedman (2001). El boosting es una estrategia de ensamblado que consiste en construir un modelo fuerte combinando secuencialmente múltiples modelos débiles, generalmente árboles de decisión de baja profundidad.

La idea central del Gradient Boosting es ajustar iterativamente nuevos modelos a los residuos del modelo anterior, de manera que cada paso corrige los errores cometidos previamente. El procedimiento general puede resumirse en los siguientes pasos:

1. Se entrena un modelo inicial (por ejemplo, un árbol de decisión poco profundo).
2. Se calculan los residuos (errores) de las predicciones respecto a los valores reales.
3. Se entrena un nuevo modelo para predecir esos residuos.
4. El nuevo modelo se combina con los anteriores mediante una tasa de aprendizaje (*learning rate*) que controla la contribución de cada estimador.
5. El proceso se repite iterativamente hasta alcanzar un número predefinido de iteraciones o hasta que el error converja.

El término “gradient” en Gradient Boosting proviene de que, en lugar de ajustar los modelos directamente a los residuos, la optimización se formula como un descenso por gradiente sobre una función de pérdida arbitraria (por ejemplo, error cuadrático medio en regresión). Esto hace que el método sea altamente flexible y aplicable a una amplia variedad de problemas de regresión y clasificación.

Las principales características del GBR son:

* Alta capacidad predictiva, incluso en escenarios con **relaciones no lineales** complejas.
* Manejo eficiente de variables heterogéneas (numéricas y categóricas).
* Posibilidad de ajustar hiperparámetros como el número de árboles, la profundidad máxima, la tasa de aprendizaje y la función de pérdida.

En aplicaciones agrícolas y climáticas, el GBR ha demostrado gran efectividad para modelar relaciones no lineales entre variables ambientales y rendimientos productivos, complementando a modelos más secuenciales como las LSTM. Por ejemplo, puede ser utilizado para estimar el rendimiento esperado de un cultivo en función de variables agregadas como temperatura, precipitación acumulada, humedad relativa o velocidad de los vientos.

# Descripción del proyecto

El actual proyecto fue desarrollado con el fin de obtener la mejor distribución del sembrado de diferentes semillas en un área específica para que el productor obtenga la mayor ganancia posible basándonos en los precios actuales del mercado.

Es por ello que en un inicio, al ejecutar el programa, se abre una pantalla emergente de una aplicación que le permite al usuario calcular los metros cuadrados y la localización de su campo. En dicha ventana emergente el usuario abrirá un GIS, éste es un mapa interactivo que permite formar un polígono a partir de la selección de varios puntos vértices en un mapa; una vez seleccionada la zona de siembra del campo, se le hace click derecho cargando las coordenadas de los diferentes puntos vértices a la pantalla emergente y cerrando el GIS. A dichas coordenadas se las procesa, obteniéndose así los metros cuadrados, el departamento al que pertenece el campo y el centroide (es el punto medio de una figura o cuerpo, que representa el promedio de la posición de todos sus puntos) del polígono formado por los puntos vértices.

Luego, con dichas coordenadas del centroide se recuperan los datos climáticos de la estación meteorológica más cercana mediante una URL de la NASA, pero se recuperan los datos diarios, por lo que los hacemos pasar por una función que calcula el promedio en lapsos de 30 días, creándose así el archivo de datos climáticos mensuales históricos en ese punto. Con este archivo es que se entrenan las redes neuronales predictoras de datos climáticos: LSTM para predecir valores de humedad y temperatura, ya que son secuenciales y mediante una ventana (valores previos a la fecha a predecir) de 14 meses interpreta como son los ciclos anuales; no así con las columnas de precipitaciones y vientos, que sus datos no siguen una secuencia (o al menos no en cuestión de meses), por lo que fueron implementados modelos de redes neuronales de GBR, las cuales son mejores que las LSTM para predecir datos que no siguen una secuencia, es decir, tienen más aleatoriedad. Una vez entrenados los modelos, estos se guardan en una carpeta llamada “model” para luego ser recuperados cuando se necesite predecir los valores climáticos futuros sin tener que pasar por un re-entreno.

A continuación, existe una validación para corroborar de que existan los archivos de entreno del GBR principal que va a predecir las toneladas que se pueden obtener de una semilla. El archivo principal del que derivan los demás es df\_semillas\_suelo\_clima.csv, en el cual se unieron los archivos de los datos del suelo promedio por departamento, los datos de las semillas cosechadas por departamento y los datos climáticos de los 14 meses previos al año en que se cosecharon las semillas en tal departamento. Luego se filtran columnas innecesarias de dicho archivo para obtener en limpio uno nuevo en el cual solo queden los datos útiles de entrada y la columna de la salida esperada para el entrenamiento del modelo.

El entrenamiento del modelo principal tiene una validación de que no se realiza si es que ya existe un modelo de este almacenado, ya que el entrenamiento de este tarda varias horas, y aún más si se hace un hyperparameter tuning. Este concepto consta de correr varios entrenamientos del modelo pero con diferentes valores de hiperparámetros, almacenando los que mejor score produzcan y poder saber con cuáles es que la red neuronal trabajará de la mejor manera; pero este trabajo, para la cantidad de datos que tenemos, puede llevar horas. Al igual que los anteriores, una vez entrenado el modelo, este se almacena para luego ser recuperado y utilizado. Cabe aclarar que como este modelo no reconoce categorías o strings, se adaptaron los nombres de las semillas a enteros mediante un diccionario, el cual también fue almacenado en “model” para que pueda ser recuperado y utilizado en el algoritmo genético cuando se necesite reemplazar el nombre de la semilla, por su correspondiente entero que fue utilizado para entrenar el modelo, para crear los datos de entradas que generen la predicción de toneladas obtenidas por hectáreas.

Por último, se ejecuta el algoritmo genético. En este se especifican las siete semillas sobre las cuales se evaluará el sembrado, ya que no se logró la obtención de los valores de mercado de fuentes confiables de las demás. Entonces, como son siete las semillas a evaluar cada individuo creado para la población es un arreglo de siete valores reales que representan el área de sembrado de cada semilla, estos siete valores son aleatorios y deben sumar en total por individuo la cantidad de hectáreas que se obtuvieron en con la aplicación de cálculo de área.

Otras modificaciones particulares que tiene este algoritmo genético por sobre los más comunes es la forma en que se realiza la mutación, como los genes de los individuos son números reales se implementa la mutación del tipo swap. Esta mutación consiste en invertir la posición de dos genes, es decir, intercambian las áreas de sembrado de dos semillas aleatorias.

En este caso particular se busca que se mantengan los individuos que su distribución áreas de sembrado por semilla genere la mayor ganancia posible al productor, es por ello, que la función objetivo es el cálculo de cuánto dinero se estima obtener por la cosecha de cada semilla. Aquí es dónde entran las redes neuronales, porque para predecir las toneladas que se obtendrán es necesario generar datos de entrada con las mismas dimensiones con el que fue entrenado, por ende, por cada individuo se crea un dataframe de entrada por semilla con los valores de entero correspondiente al nombre de la semilla, datos del suelo del departamento al que pertenece el campo, supuesta superficie sembrada y los datos climáticos de los próximos catorce meses predichos por las redes neuronales previamente mencionadas para el clima. Con dichos datos de entrada la red neuronal principal devuelve las toneladas, generando un arreglo en el que cada posición corresponde a las toneladas obtenidas para cada semilla. Este arreglo es recorrido y multiplicado por el precio por tonelada de cada semilla, lo que sumándolos se obtiene la ganancia total esperada para la producción con esa distribución de áreas de sembrado para cada semilla, ese valor total es nuestro valor objetivo que se busca maximizar con las diferentes corridas del AG.

Al finalizar la cantidad de corridas especificadas, se le muestra al usuario un gráfico de la evolución de la población y un gráfico de la distribución de áreas a sembrar por semilla correspondiente al mejor individuo obtenido luego de la ultima iteración.