



TRABAJO DE GRADO
Opción Seminario-Diplomado.

**ALGORITMO COMPUTACIONAL PARA EL ANÁLISIS Y TOMA DE DECISIONES
EN LA CLASIFICACIÓN DE LOS DATOS SOBRE EL CRECIMIENTO DE LAS
PLANTAS, UTILIZANDO ESTRATEGIAS DE MACHINE LEARNING**

Corporación Universitaria Remington.
Nombre de la facultad: Ingenierías
Nombre del programa académico: Ingeniería En Sistemas

Estudiantes:
Juan Nicolas Rodriguez Salazar.
Tutor: Juan Carlos Briñez de León
Opción de Trabajo de grado Seminario-Diplomado.
2024.

Dedicatoria

A mis padres, por todo el apoyo y motivación para ser un profesional, quienes han estado en esta batalla académica, y por quienes me han ayudado.

Agradecimientos

A DIOS, A LA UNIVERSIDAD, A MIS PROFESORES, A MIS COMPAÑEROS, ETC.

agradezco a todos los que han estado por el apoyo recibido en mi trabajo de grado.

Tabla de Contenidos

| | |
|--|----|
| Resumen | 5 |
| Palabras clave | 5 |
| Marco conceptual y contextual | 6 |
| Pregunta problema | 7 |
| Descripción de variables | 8 |
| Aproximaciones con gráficos - analítica | 13 |
| Objetivos | 17 |
| Objetivo general | 17 |
| Objetivos específicos | 17 |
| Desarrollo e implementación de aprendizaje | 18 |
| Validación del modelo | 25 |
| Conclusiones y trabajos futuros | 28 |
| Referencias bibliográficas | 29 |

Resumen

En este conjunto de datos la tarea de predicción normalmente implicaría predecir o clasificar el hito de crecimiento de las plantas en función de los factores ambientales y de gestión proporcionados. En concreto, se intentará predecir la etapa o el hito de crecimiento que alcanza una planta en función de variables como el tipo de suelo, las horas de luz solar, la frecuencia de riego, el tipo de fertilizante, la temperatura y la humedad. Esta predicción puede ayudar a comprender cómo las diferentes condiciones influyen en el crecimiento de las plantas y puede ser valiosa para optimizar las prácticas agrícolas o la gestión de invernaderos.

Palabras clave

Plantas, crecimiento, clasificación, suelo, fertilizante, frecuencia de riego luz solar, temperatura y humedad.

Marco conceptual y contextual

Esta investigación identifica y analiza la clasificación y el crecimiento de las plantas, mostrando cada una de sus etapas en función de variables como el tipo de suelo, las horas de luz solar, la frecuencia de riego, el tipo de fertilizante, la temperatura y la humedad para saber el estado de las plantas, si la planta crecerá bien o no crecerá.

Estos datos ayudan a comprender cómo las condiciones del clima influyen en el crecimiento de las plantas, optimizando las prácticas agrícolas y la gestión del campo.

Esta recopilación de datos nos ayudara mucho a como cultivar las plantas determinando su tipo de suelo, que tipo de fertilizante podemos echarle, cuánta agua necesita y, además, podemos reforestar zonas que nos den sombra para mejorar nuestra calidad de vida.

Pregunta problema:

¿Qué características hacen que las plantas crezcan de forma eficiente? ¿Qué beneficios nos dan?

Análisis: Realizar un análisis de regresión logística para evaluar la relación entre tipos de humedad, tipo de suelo y horas de sol y cómo estos afectan a las plantas al crecer.

¿Cuál es la distribución por horas de sol adecuada? ¿Hay una tendencia específica en la cantidad de agua proporcionada?

Análisis: Realizar un análisis de distribución de frecuencias para evaluar la distribución de agua y horas de sol adecuadas.

Acercamiento a los datos: La información usada para este proyecto proviene de un dataset en la página de Kaggle y expone variables o características que afectan a las plantas al momento de crecer, dicho dataset está bajo el nombre de “Plant Growth Data Clasification”.

Descripción de variables.

| | Soil_Type | Sunlight_Hours | Water_Frequency | Fertilizer_Type | Temperature | Humidity | Growth_Milestone |
|----|-----------|------------------------|-----------------|-----------------|------------------------|------------------------|------------------|
| 0 | loam | 5.192.294.089.205.030 | bi-weekly | chemical | 31.719.602.410.244.100 | 6.159.186.060.848.990 | 0 |
| 1 | sandy | 4.033.132.702.741.610 | weekly | organic | 2.891.948.412.187.390 | 5.242.227.609.891.590 | 1 |
| 2 | loam | 8.892.768.570.729.000 | bi-weekly | none | 23.179.058.888.285.300 | 4.466.053.858.490.320 | 0 |
| 3 | loam | 8.241.144.063.085.700 | bi-weekly | none | 18.465.886.401.416.900 | 464.332.272.684.958 | 0 |
| 4 | sandy | 8.374.043.008.245.920 | bi-weekly | organic | 1.812.874.085.342.170 | 6.362.592.280.385.190 | 0 |
| 5 | sandy | 8.627.622.080.115.670 | bi-weekly | none | 20.004.857.963.291.900 | 67.618.726.471.884 | 0 |
| 6 | loam | 4.444.267.910.404.540 | daily | organic | 25.984.533.294.122.400 | 6.957.895.218.629.240 | 1 |
| 7 | clay | 6.150.794.371.265.630 | daily | organic | 29.291.918.454.001.200 | 6.948.090.713.972.760 | 0 |
| 8 | loam | 4.695.214.357.150.770 | bi-weekly | none | 28.203.947.534.354.600 | 34.560.305.152.434.500 | 1 |
| 9 | loam | 9.178.620.555.253.560 | weekly | organic | 20.598.677.938.918.800 | 5.472.101.523.512.900 | 1 |
| 10 | loam | 77.397.887.609.653.400 | daily | none | 34.097.305.613.263.800 | 32.877.938.000.832.200 | 0 |
| 11 | loam | 5.985.388.149.115.890 | bi-weekly | chemical | 2.975.793.833.391.530 | 5.747.644.411.618.670 | 0 |

Figura 1: tabla de valores del Dataset

Aquí se evidencia la tabla con los valores, donde se encuentran características como el tipo de suelo, horas de luz solar, la frecuencia de riego, el tipo de fertilizante, la temperatura ambiente, la humedad y finalmente si el crecimiento de la planta fue positivo confirmándolo con un 1 o negándolo con un 0.

Mediante un análisis de variables se evidencia que el dataset cuenta con 7 columnas de variables, de las cuales 6 son de tipo objeto para texto y números decimales y solamente 1 es de tipo entero, siendo esta el crecimiento de la planta.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 193 entries, 0 to 192
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Soil_Type              193 non-null    object
1   Sunlight_Hours        193 non-null    object
2   Water_Frequency       193 non-null    object
3   Fertilizer_Type       193 non-null    object
4   Temperature           193 non-null    object
5   Humidity              193 non-null    object
6   Growth_Milestone      193 non-null    int64
dtypes: int64(1), object(6)
memory usage: 10.7+ KB
```

Figura 2: analizando los tipos de variables del Dataset

Haciendo un análisis estadístico rápido de los datos se puede observar cosas como que la media de humedad es de 59.18 así como que la temperatura jamás sobrepasa los 40 grados y que el mínimo de horas de luz solar que una planta ha recibido en la recopilación de estos datos ha sido de 4 horas, un tiempo muy menor para una planta.

```
#Análisis estadístico rápido de los datos
Conjunto_Datos.describe()
```

| | Sunlight_Hours | Temperature | Humidity | Growth_Milestone |
|-------|----------------|-------------|------------|------------------|
| count | 193.000000 | 193.000000 | 193.000000 | 193.000000 |
| mean | 6.826484 | 25.076087 | 58.098927 | 0.497409 |
| std | 1.599509 | 5.354170 | 12.631799 | 0.501294 |
| min | 4.033133 | 15.200000 | 30.567682 | 0.000000 |
| 25% | 5.477000 | 20.637095 | 49.300000 | 0.000000 |
| 50% | 6.833290 | 25.912336 | 59.182806 | 0.000000 |
| 75% | 8.241144 | 29.757938 | 69.100000 | 1.000000 |
| max | 9.913903 | 34.810103 | 79.648240 | 1.000000 |

Figura 3: analizando estadísticamente los datos del Dataset

Verificando las variables del Dataset se ve que en la columna de frecuencia de riego son: quincenal, semanal y diariamente.

```
#Verificación de las opciones de la variable
print('Analizando la decisión')
Datos_Loan['Water_Frequency'].unique()

Analizando la decisión
array(['bi-weekly', 'weekly', 'daily'], dtype=object)
```

Figura 4: Verificando la variable frecuencia de riego (diario, semanal y quincenal)

Analizando las variables del tipo de suelo se encuentran: arenoso, barro o tierra húmeda y finalmente marga que es un tipo de tierra sin arena o arcilla.

Este análisis nos indica el tipo de suelo apropiado para cada cultivo que se va a sembrar.

```
#Analizando variables categóricas
Conjunto_Datos['Soil_Type'].unique()

array(['loam', 'sandy', 'clay'], dtype=object)
```

Figura 5: Analizando la variable tipo suelo (suelo margoso, suelo arenoso y suelo arcilloso)

Finalmente observando las variables de la columna de tipo de fertilizante se encuentran: fertilizante químico, fertilizante orgánico y ninguno

```
#Analizando variables categóricas
Conjunto_Datos['Fertilizer_Type'].unique()

array(['chemical', 'organic', 'none'], dtype=object)
```

Figura 6: analizando las variables de la columna tipo de fertilizante (fertilizante químico, fertilizante orgánico y ninguno)

Aproximaciones con gráficos - analítica.

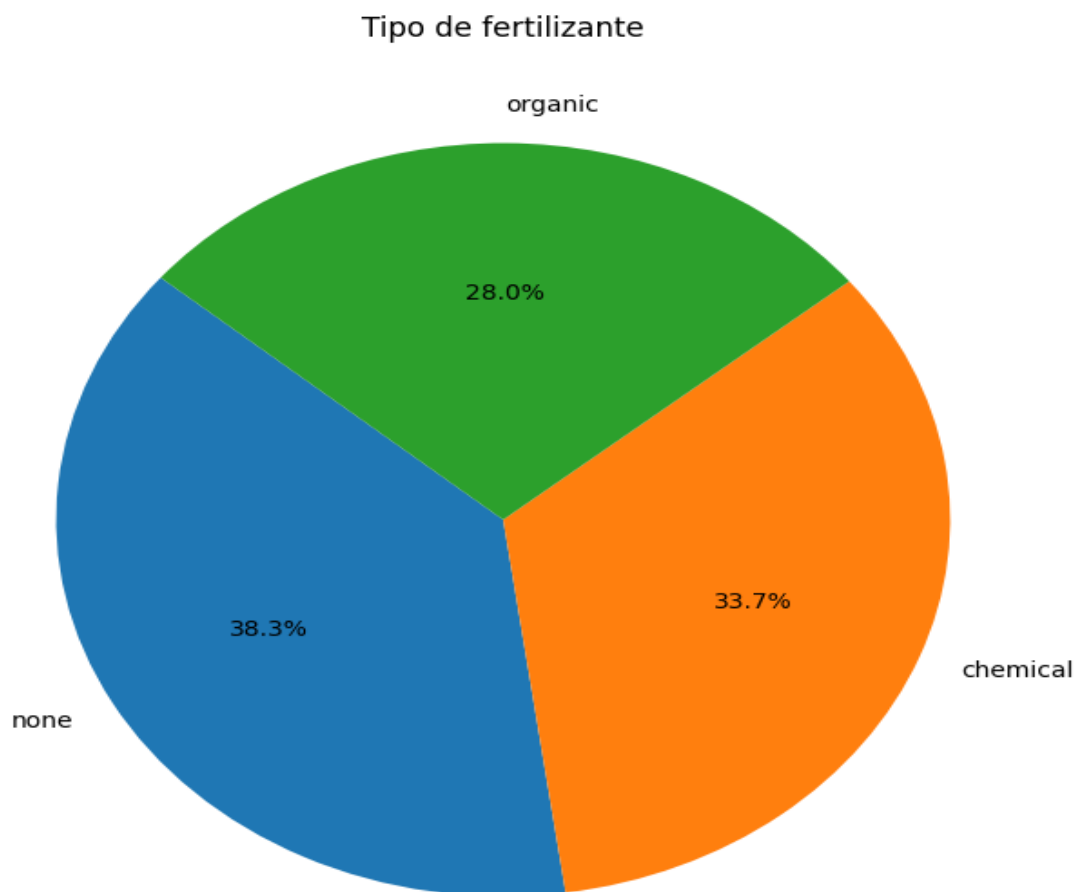


Figura 7: grafica de torta 1 explicando el tipo de fertilizante que se ha usado por medio del porcentaje

En este grafico de torta explica el porcentaje del tipo de fertilizante que se ha usado

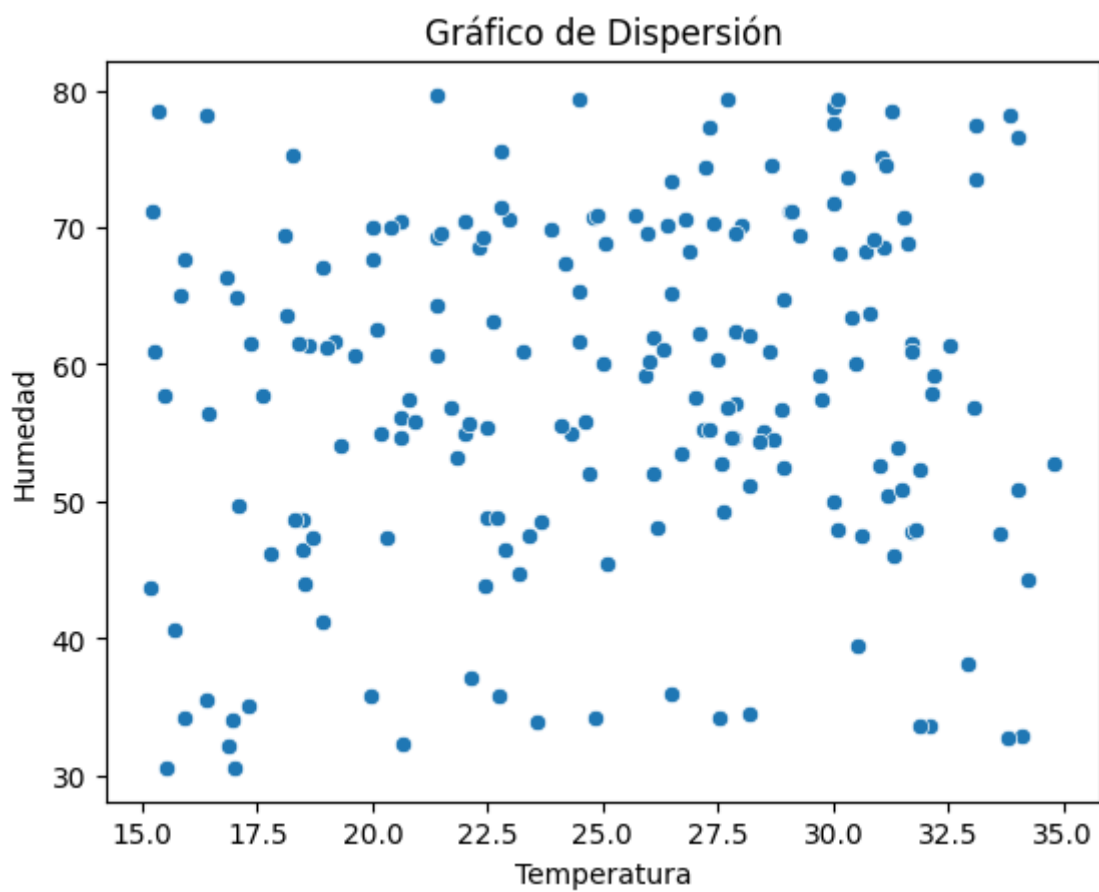


Figura 8: Gráfica de dispersión indicando el nivel de temperatura y humedad del suelo

En este gráfico de dispersión nos indica el nivel de temperatura y humedad del suelo.

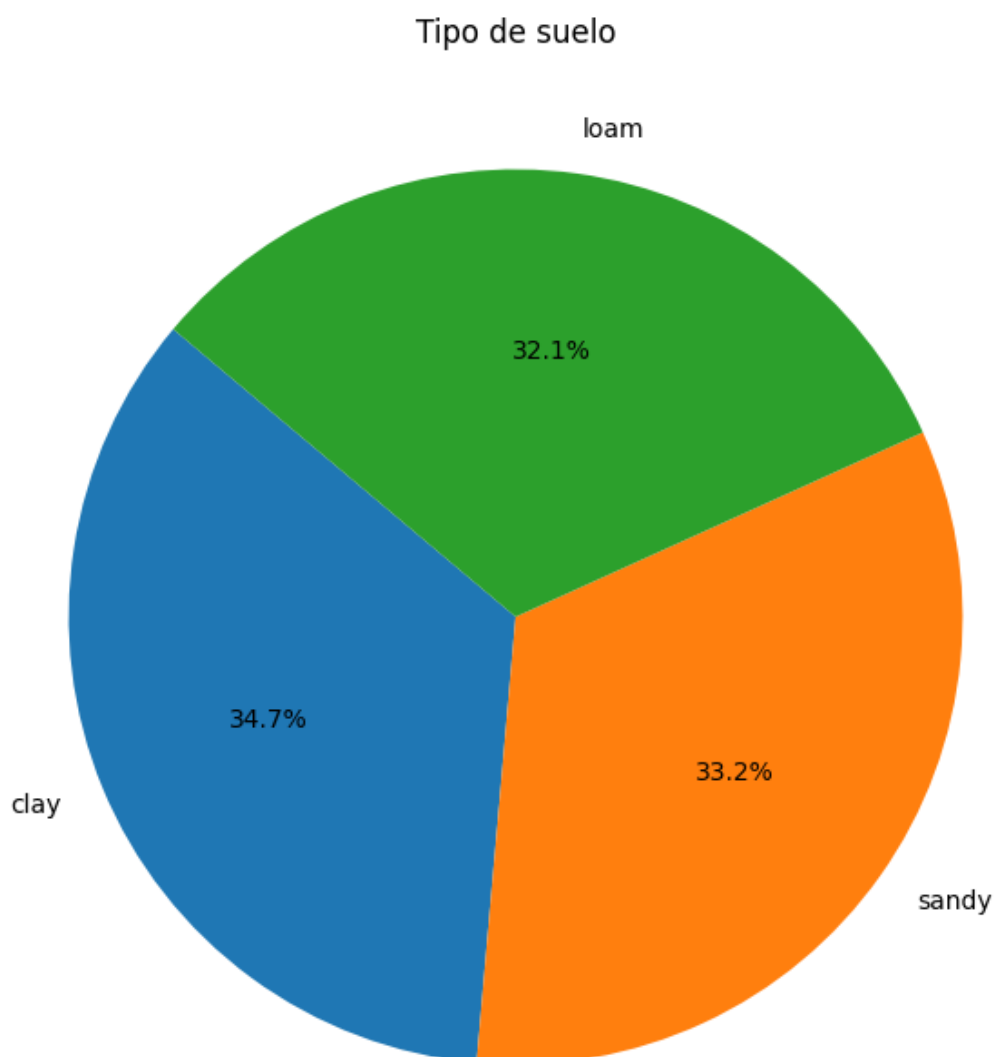


Figura 9: Gráfico de torta 2 explicando el tipo de suelo que hemos usado para cultivar las plantas por medio del porcentaje

En esta Segunda gráfica de torta, nos dice cuanto tipo de suelo hemos usado para cultivar las plantas mediante porcentaje.

En esta figura colocamos números que representan el tipo de suelo, frecuencia de riego de las plantas y el tipo de fertilizante para que nos arrojen datos de grados acerca de las horas de la luz solar, la temperatura y humedad.

```
#Mapeando todas la variables categóricas a numéricas
Reemplazo_1={'loam':30,'sandy':60,'clay':90}
Datos_Loan['Soil_Type']=Datos_Loan['Soil_Type'].map(Reemplazo_1)

Reemplazo_2={'chemical':30,'organic':60,'none':90}
Datos_Loan['Fertilizer_Type']=Datos_Loan['Fertilizer_Type'].map(Reemplazo_2)

Reemplazo_3={'bi-weekly':30,'weekly':60,'daily':90}
Datos_Loan['Water_Frequency']=Datos_Loan['Water_Frequency'].map(Reemplazo_3)

Datos_Loan.head()
```

| | Soil_Type | Sunlight_Hours | Water_Frequency | Fertilizer_Type | Temperature | Humidity | Growth_Milestone |
|---|-----------|----------------|-----------------|-----------------|-------------|-----------|------------------|
| 0 | 30 | 5.192294 | 30 | 30 | 31.719602 | 61.591861 | 0 |
| 1 | 60 | 4.033133 | 60 | 60 | 28.919484 | 52.422276 | 1 |
| 2 | 30 | 8.892769 | 30 | 90 | 23.179059 | 44.660539 | 0 |
| 3 | 30 | 8.241144 | 30 | 90 | 18.465886 | 46.433227 | 0 |
| 4 | 60 | 8.374043 | 30 | 60 | 18.128741 | 63.625923 | 0 |

Figura 10: mapeando todas las variables categóricas a numéricas

Objetivos

Objetivo general: analizar los datos sobre el crecimiento y clasificación de las plantas de acuerdo al tipo de suelo, temperatura y nivel de riego.

Objetivos específicos:

- Caracterizar y procesar los datos de interés, con miras a la toma de decisiones informadas
- Implementar un algoritmo de Machine learning para la toma de decisiones a partir de los datos de interés.
- Evaluar y analizar el desempeño de los algoritmos implementados para la toma de decisiones.
- Validar el funcionamiento de toma de decisiones a partir de datos nuevos.

Desarrollo e implementación del aprendizaje

Gracias a estos resultados, podemos determinar: que tipo de suelo estamos usando, cuantas horas de luz solar usamos, cuánta agua le estamos echando a las plantas, que tipo de fertilizante estamos usando, cual es el nivel de temperatura, cuantos grados de humedad hay y que hito de crecimiento indica el crecimiento de una planta.

Preparación de los datos

Todo lo que hicimos en las primeras clases

En esta figura colocamos números que representan el tipo de suelo, frecuencia de riego de las plantas y el tipo de fertilizante para que nos arrojen datos de grados acerca de las horas de la luz solar, la temperatura y humedad.

```
#Mapeando todas la variables categóricas a numéricas
Reemplazo_1={'loam':30,'sandy':60,'clay':90}
Datos_Loan['Soil_Type']=Datos_Loan['Soil_Type'].map(Reemplazo_1)

Reemplazo_2={'chemical':30,'organic':60,'none':90}
Datos_Loan['Fertilizer_Type']=Datos_Loan['Fertilizer_Type'].map(Reemplazo_2)

Reemplazo_3={'bi-weekly':30,'weekly':60,'daily':90}
Datos_Loan['Water_Frequency']=Datos_Loan['Water_Frequency'].map(Reemplazo_3)

Datos_Loan.head()
```

| | Soil_Type | Sunlight_Hours | Water_Frequency | Fertilizer_Type | Temperature | Humidity | Growth_Milestone |
|---|-----------|----------------|-----------------|-----------------|-------------|-----------|------------------|
| 0 | 30 | 5.192294 | 30 | 30 | 31.719602 | 61.591861 | 0 |
| 1 | 60 | 4.033133 | 60 | 60 | 28.919484 | 52.422276 | 1 |
| 2 | 30 | 8.892769 | 30 | 90 | 23.179059 | 44.660539 | 0 |
| 3 | 30 | 8.241144 | 30 | 90 | 18.465886 | 46.433227 | 0 |
| 4 | 60 | 8.374043 | 30 | 60 | 18.128741 | 63.625923 | 0 |

Figura 11: mapeando todas las variables categóricas a numéricas

Modelo de toma de decisiones

Puede ser clustering o lo que hace falta por ver

```
#Divide datos en entradas y salidas
import numpy as np
Datos_matriz=np.array(Datos_Loan)
#Datos_matriz[np.isnan(Datos_matriz)] = 0
X = Datos_matriz[:,0:6] #datos de entrada (Todas las variables de las encuestadas)
Y = Datos_matriz[:, -1] #Datos de salida (Presencia de violencia domestica)
print('Proceso realizado')
```

Proceso realizado

```
# Divide datos en Entrenamiento y testeo
import sklearn
from sklearn.model_selection import train_test_split
X_train, X_test,Y_train, Y_test= train_test_split(X,Y,test_size=0.1,random_state=751)
print('Proceso realizado')
```

Proceso realizado

Figura 12: dividiendo los datos en entradas y salidas para después entrenarlos y ponerlos a prueba

```
#Para mejorar la escala de los datos se hace normalization (Ignorar)
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Figura 13: mejorando la escala de los datos para transformarlos muy bien

```

# Evaluando casos mediante todos los clasificadores
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score,precision_score

Modelo_0 = KNeighborsClassifier(5)
Modelo_0.fit(X_train, Y_train)
Y_pred_0 =Modelo_0.predict (X_test)
print("Accuracy KNN",accuracy_score(Y_test, Y_pred_0))

Modelo_1 = GaussianNB()
Modelo_1.fit(X_train, Y_train)
Y_pred =Modelo_1.predict (X_test)
print("Accuracy Bayes",accuracy_score(Y_test, Y_pred))

Modelo_2 = LinearDiscriminantAnalysis()
Modelo_2.fit(X_train, Y_train)
Y_pred_2 =Modelo_2.predict (X_test)
print("Accuracy LDA",accuracy_score(Y_test, Y_pred_2))

Modelo_3 = QuadraticDiscriminantAnalysis()
Modelo_3.fit(X_train, Y_train)
Y_pred_3 =Modelo_3.predict (X_test)
print("Accuracy QDA",accuracy_score(Y_test, Y_pred_3))

```

Figura 14: evaluando casos mediante clasificadores

```

Modelo_4 = DecisionTreeClassifier()
Modelo_4.fit(X_train, Y_train)
Y_pred_4 =Modelo_4.predict (X_test)
print("Accuracy Tree",accuracy_score(Y_test, Y_pred_4))

Modelo_5 = SVC()
Modelo_5.fit(X_train, Y_train)
Y_pred_5 =Modelo_5.predict (X_test)
print("Accuracy SVM",accuracy_score(Y_test, Y_pred_5))

Accuracy KNN 0.5
Accuracy Bayes 0.5
Accuracy LDA 0.55
Accuracy QDA 0.65
Accuracy Tree 0.5
Accuracy SVM 0.7

```

Figura 15: nos muestra estos resultados

```

import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, roc_curve, auc, classification_report
import seaborn as sns

# Calcular métricas
accuracy = Modelo_4.score(X_test, Y_test)
cm = confusion_matrix(Y_test, Y_pred_0)
report = classification_report(Y_test, Y_pred_0)

# Calcular la curva ROC y el área bajo la curva (AUC)
Y_pred_prob = Modelo_4.predict_proba(X_test)[:, 1]
fpr, tpr, _ = roc_curve(Y_test, Y_pred_prob)
roc_auc = auc(fpr, tpr)

# Mostrar resultados
print(f"Accuracy: {accuracy:.2f}")
print('=====')
print(' ')

# Visualizar la matriz de confusión
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix')
plt.show()
print('=====')

```

Figura 16: calculando las bayas métricas, la curva ROC y el área bajo la curva para que nos genere un reporte de cultivo de una planta

```

print('=====')
print(' ')

# Visualizar la matriz de confusión
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix')
plt.show()
print('=====')

print(' ')
print("Classification Report:")
print(report)
print('=====')

# Visualizar la curva ROC
print(' ')
plt.figure()
lw = 2
plt.plot(fpr, tpr, color='darkorange', lw=lw, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve')
plt.legend(loc="lower right")
plt.show()

```

Figura 17: visualizando la matriz de confusión para ejecutar un reporte en la figura 19 y visualizando la curva ROC que se mostrara en la figura 20

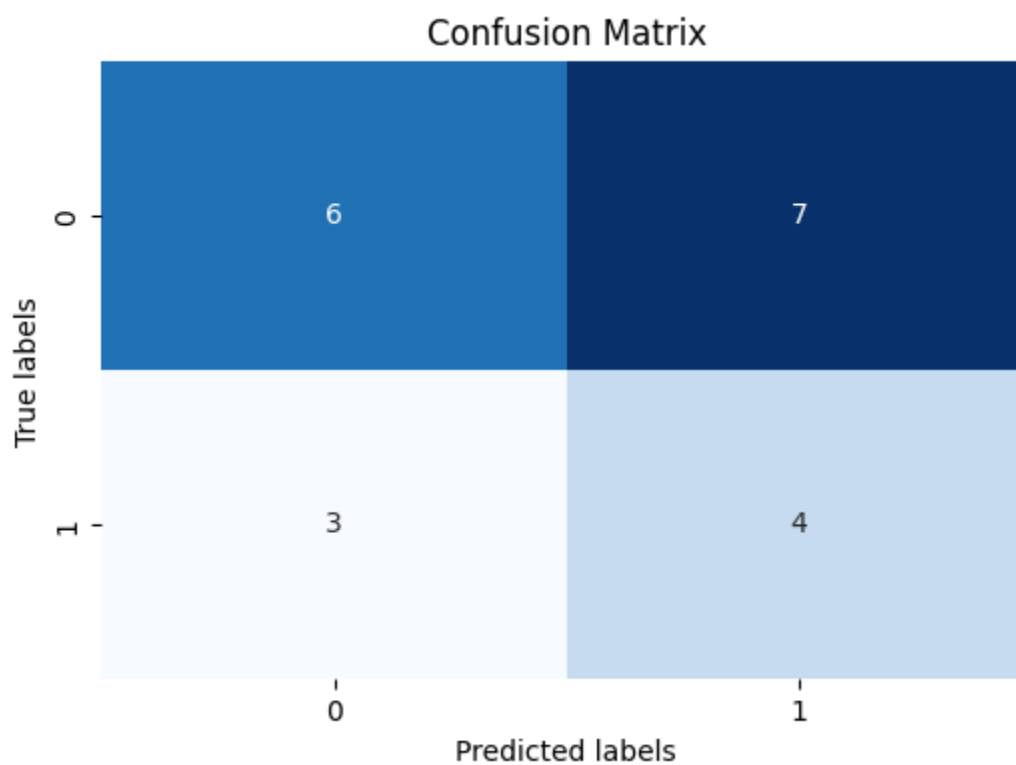


Figura 18: esta matriz de confusión nos indica cuantas etiquetas estamos prediciendo

```

=====
Classification Report:
              precision    recall  f1-score   support

     0.0         0.67         0.46         0.55         13
     1.0         0.36         0.57         0.44          7

 accuracy          0.50         20
 macro avg         0.52         0.52         0.49         20
 weighted avg      0.56         0.50         0.51         20
=====

```

Figura 19: reporte de bayas métricas

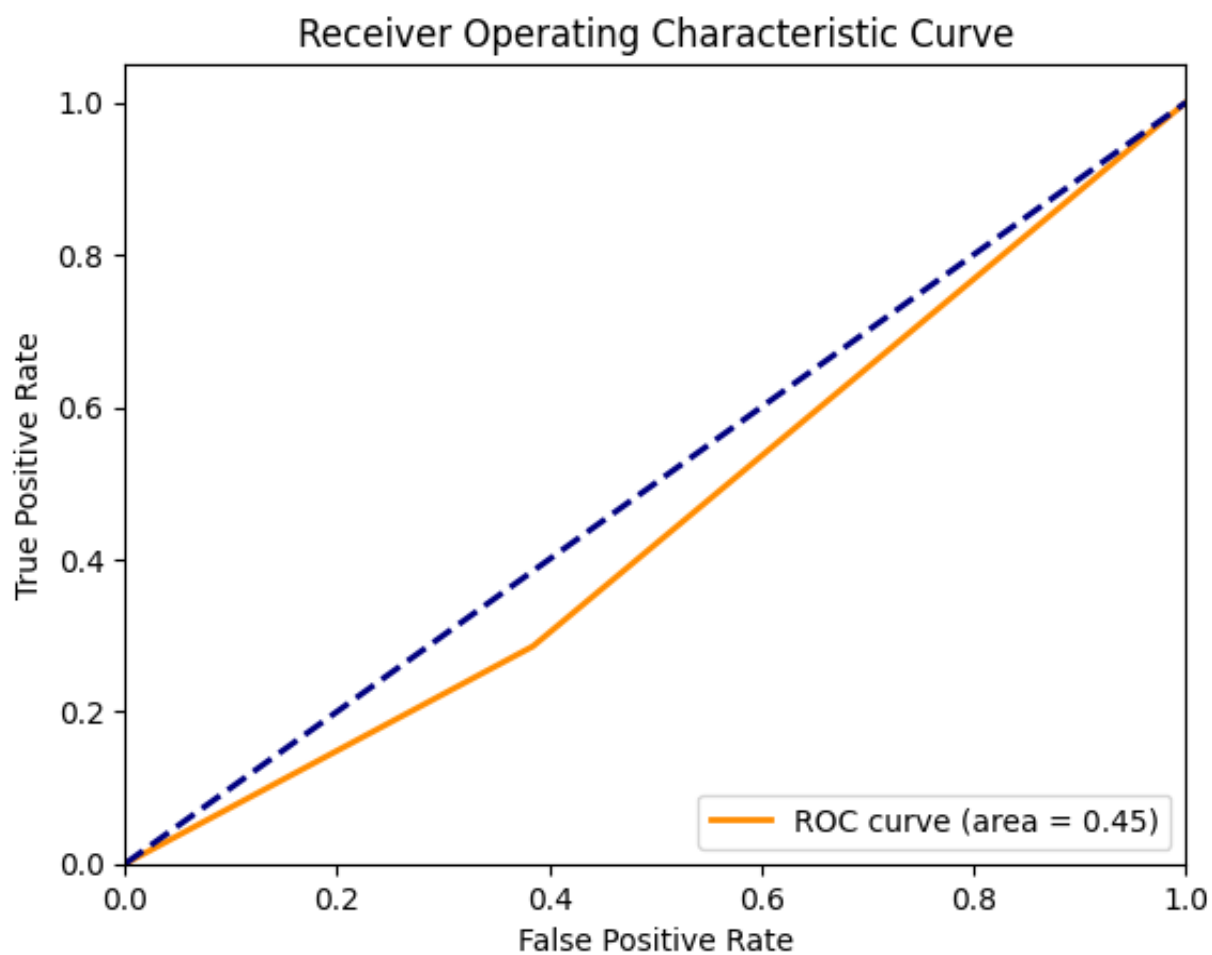


Figura 20: Gráfica de curva Roc que nos muestra el área

Validación del modelo

A continuación, se mostrarán 5 ejercicios probando los diferentes modelos de IA:

Probamos el modelo entrenado para digitar el nivel de las variables para ver si el programa nos dice si crecerá bien o no crecerá bien.

```
#Probando el modelo entrenado sobre un nuevo sujeto
Target=np.zeros((1,6))
Target[0,0]=float(input('Ingrese tipo de suelo, 30 marga, 60 arenoso, 90 barro: '))
Target[0,1]=float(input('Ingrese horas de luz, entre 4 y 10: '))
Target[0,2]=float(input('Ingrese frecuencia de riego, 30 quincenal, 60 semanal, 90 diariamente: '))
Target[0,3]=float(input('Ingrese tipo de fertilizante, 30 químico, 60 orgánico, 90 ninguno: '))
Target[0,4]=float(input('Ingrese temperatura, entre 15 y 35 grados: '))
Target[0,5]=float(input('Ingrese humedad, entre 30 y 80: '))

Target = scaler.transform(Target) #Normalizar los datos

Prediction_0 =Modelo_0.predict (Target)
Prediction_1 =Modelo_1.predict (Target)
Prediction_2 =Modelo_2.predict (Target)
Prediction_3 =Modelo_3.predict (Target)
Prediction_4 =Modelo_4.predict (Target)
Prediction_5 =Modelo_5.predict (Target)

print(" ")

if Prediction_0==0:
    print("Según KNN, no crecerá bien")
else:
    print("Según KNN, crecerá bien")

print(" ")
```

Figura 21: probando el modelo entrenado para digitar el nivel de las variables para ver si el programa nos dice si crecerá bien o no crecerá bien según los programas.

Vamos a predecir 5 casos, después de digitar los 5 casos, el programa nos dirá si la planta crecerá bien o no crecerá bien según los Metodos aplicados.

```
if Prediction_1==0:
    print("Según Bayes, no crecerá bien")
else:
    print("Según Bayes, crecerá bien")

print(" ")

if Prediction_2==0:
    print("Según LDA, no crecerá bien")
else:
    print("Según LDA, crecerá bien")

print(" ")

if Prediction_3==0:
    print("Según QDA, no crecerá bien")
else:
    print("Según QDA, crecerá bien")

print(" ")

if Prediction_4==0:
    print("Según Tree, no crecerá bien")
else:
    print("Según tree, crecerá bien")

print(" ")
```

Figura 22: realizando 5 predicciones para ver si el programa nos dirá si la planta crecerá bien o no crecerá bien según cada uno de los métodos.

Después de digitar los niveles estadísticos en las variables, se nos mostraran 5 resultados de parte de los Metodos, y todos nos arrojan resultados positivos.

```
if Prediction_5==0:
    print("Según SVM, no crecerá bien")
else:
    print("Según SVM, crecerá bien")

print(" ")

Ingrese tipo de suelo, 30 marga, 60 arenoso, 90 barro: 60
Ingrese horas de luz, entre 4 y 10: 5
Ingrese frecuencia de riego, 30 quincenal, 60 semanal, 90 diariamente: 60
Ingrese tipo de fertilizante, 30 químico, 60 orgánico, 90 ninguno: 30
Ingrese temperatura, entre 15 y 35 grados: 20
Ingrese humedad, entre 30 y 80: 40

Según KNN, crecerá bien

Según Bayes, crecerá bien

Según LDA, crecerá bien

Según QDA, crecerá bien

Según tree, crecerá bien

Según SVM, crecerá bien
```

Figura 23: mostrando resultados de las predicciones de las variables

Tabla 1. Clasificación de datos sobre el crecimiento de las plantas

| | A | B | C | D | E | F | G |
|----|-----------|------------------------|-----------------|-----------------|------------------------|------------------------|------------------|
| 1 | Soil_Type | Sunlight_Hours | Water_Frequency | Fertilizer_Type | Temperature | Humidity | Growth_Milestone |
| 2 | loam | 5.192.294.089.205.030 | bi-weekly | chemical | 31.719.602.410.244.100 | 6.159.186.060.848.990 | 0 |
| 3 | sandy | 4.033.132.702.741.610 | weekly | organic | 2.891.948.412.187.390 | 5.242.227.609.891.590 | 1 |
| 4 | loam | 8.892.768.570.729.000 | bi-weekly | none | 23.179.058.888.285.300 | 4.466.053.858.490.320 | 0 |
| 5 | loam | 8.241.144.063.085.700 | bi-weekly | none | 18.465.886.401.416.900 | 464.332.272.684.958 | 0 |
| 6 | sandy | 8.374.043.008.245.920 | bi-weekly | organic | 1.812.874.085.342.170 | 6.362.592.280.385.190 | 0 |
| 7 | sandy | 8.627.622.080.115.670 | bi-weekly | none | 20.004.857.963.291.900 | 67.618.726.471.884 | 0 |
| 8 | loam | 4.444.267.910.404.540 | daily | organic | 25.984.533.294.122.400 | 6.957.895.218.629.240 | 1 |
| 9 | clay | 6.150.794.371.265.630 | daily | organic | 29.291.918.454.001.200 | 6.948.090.713.972.760 | 0 |
| 10 | loam | 4.695.214.357.150.770 | bi-weekly | none | 28.203.947.534.354.600 | 34.560.305.152.434.500 | 1 |
| 11 | loam | 9.178.620.555.253.560 | weekly | organic | 20.598.677.938.918.800 | 5.472.101.523.512.900 | 1 |
| 12 | loam | 77.397.887.609.653.400 | daily | none | 34.097.305.613.263.800 | 32.877.938.000.832.200 | 0 |
| 13 | loam | 5.985.388.149.115.890 | bi-weekly | chemical | 2.975.793.833.391.530 | 5.747.644.411.618.670 | 0 |
| 14 | sandy | 4.381.350.101.716.140 | daily | organic | 26.087.081.050.228.000 | 5.207.652.506.866.880 | 1 |
| 15 | loam | 5.865.893.930.293.970 | daily | chemical | 27.234.414.924.687.000 | 7.438.520.913.791.490 | 1 |
| 16 | clay | 5.951.099.932.160.480 | weekly | none | 233.920.012.485.558 | 4.754.575.062.760.390 | 1 |
| 17 | sandy | 8.377.637.070.028.380 | bi-weekly | chemical | 1.995.461.979.002.310 | 35.853.350.821.380.200 | 1 |
| 18 | clay | 78.253.448.281.312.700 | bi-weekly | chemical | 22.119.453.573.025.200 | 37.149.584.102.641.700 | 0 |
| 19 | clay | 932.327.645.545.796 | weekly | none | 3.015.692.220.928.730 | 6.807.553.158.587.360 | 0 |
| 20 | clay | 68.332.895.509.716.900 | weekly | chemical | 15.287.869.772.595.100 | 6.091.090.316.581.300 | 1 |
| 21 | clay | 471.756.547.562.981 | bi-weekly | chemical | 17.321.452.810.138.300 | 3.505.613.380.613.950 | 1 |
| 22 | sandy | 827.946.872.333.797 | bi-weekly | none | 15.920.052.840.435.000 | 3.420.534.030.574.980 | 0 |
| 23 | sandy | 8.564.710.291.701.380 | weekly | chemical | 15.814.576.046.379.400 | 65.048.456.572.956 | 1 |

Figura 24: Datos de las variables

Conclusiones y trabajos futuros

Este análisis sobre el crecimiento y la predicción de este Dataset sobre de las plantas es de gran importancia para las prácticas agrícolas dirigida a los agricultores, ingenieros forestales e ingenieros agrónomos. Con estos datos clasificados, podemos sembrar adecuadamente plantas de acuerdo con su tipo de suelo, nivel de agua, tipos de fertilizantes, nivel de temperatura y humedad, y así obtener productos de buena calidad que conlleva a beneficios económicos, reduciendo el riesgo de pérdidas y tiempo.

Referencias

Fischer, G. (2000). Crecimiento y desarrollo. *Producción, poscosecha y exportación de la uchuva*, 9-26.

DI BENEDETTO, A y TOGNETTI, J . Técnicas de análisis de crecimiento de plantas: su aplicación a cultivos intensivos. *RIA. Rev. investigando. agropecu.* [en línea]. 2016, vol.42, n.3, pp.258-282. ISSN 1669-2314.

Rivera, E., Sánchez, M., & Domínguez, H. (2018). pH como factor de crecimiento en plantas. *Revista De Iniciación Científica*, 4, 101-105. <https://doi.org/10.33412/rev-ric.v4.0.1829>

DOMINGUEZ, Jorge; LAZCANO, Cristina y GOMEZ-BRANDON, María. Influencia del vermicompost en el crecimiento de las plantas: Aportes para la elaboración de un concepto objetivo. *Acta Zool. Mex* [online]. 2010, vol.26, n.spe2, pp.359-371. ISSN 2448-8445.

Barrera, J., Suárez, D., & Melgarejo, L. M. (2010). II. Análisis de crecimiento en plantas. *Experimentos en fisiología vegetal. Melgarejo, LM (Ed). Laboratorio de Fisiología y Bioquímica Vegetal. Departamento de Biología. Universidad Nacional de Colombia*, 25-39.

<https://www.kaggle.com/datasets/gorororororo23/plant-growth-data-classification/data>