|  |
| --- |
| Título |
| Clasificación de mangos mediante técnicas de aprendizaje |
| Resumen |
| TBD |

|  |
| --- |
| Autores |
| Daniel Vidal Soroa |
| Juan Diego Peña N |
| 19/03/2022 |

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Descripción | Elaborado | Aprobado |
| 01 | Primera edición |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice

[1. Introducción 3](#_Toc98604448)

[1.1. Objeto 3](#_Toc98604449)

[1.2. Alcance 3](#_Toc98604450)

[1.3. Normas y referencias 3](#_Toc98604451)

[1.4. Definiciones y abreviaturas 3](#_Toc98604452)

[2. Descripción general 4](#_Toc98604453)

[3. Análisis de resultados 4](#_Toc98604454)

[3.1. Evaluación de modelos 4](#_Toc98604455)

[3.2. Análisis de hiperparametros 4](#_Toc98604456)

[3.3. Comparación de modelos y selección 4](#_Toc98604457)

[4. Conclusiones 4](#_Toc98604458)

# Introducción

## Objeto

El objetivo de este proyecto es implementar técnicas de aprendizaje automático supervisado sobre un dataset definido. Cada método se evaluará escogiendo las métricas de error mas adecuadas, para luego optimizar sus hiperparametros y determinar el de mayor rendimiento.

## Alcance

El proyecto incluye la aplicabilidad de modelos ya existentes en las librerías scikitlearn [1] y keras [2]. Así como el uso datos de dominio público, el proyecto no contempla el uso de técnicas de aprendizaje no supervisado y se ceñirá únicamente a las técnicas: LR, KNN, RF y CNN.

## Normas y referencias

|  |  |
| --- | --- |
| [1] | «Scikit learn,» [En línea]. Available: https://scikit-learn.org/stable/. |
| [2] | «TensorFlow,» [En línea]. Available: https://www.tensorflow.org/guide/keras?hl=es-419. |

## Definiciones y abreviaturas

**ML** Machine learning (*Aprendizaje automatico)*

**LR** Logistic regression (*Regresion logistica)*

***KNN*** K-Nearest Neighbor (K-Vecinos)

***RF*** *Random Forest (Arbol aleatorio)*

***CNN*** Convolutional neural networks (Redes neuronales convucionales

# Descripción general

El proyecto se motiva en servir como una herramienta base para países productores de mango al eliminar un control de calidad manual que posibilita el daño de la fruta y abarata el costo de otros métodos de control como el uso resonancias magnéticas. Además, debido a la migración rural negativa y el corto período de conservación de esta fruta, cada año se pueden pierden millones de toneladas debido a falta de mano de obra y errores de clasificación.

Los datos se seleccionaron de la plataforma kaggle [3] el cual cuenta con tres clases de mango y 200 imágenes por clase:

* ***Clase extra:*** Usados generalmente para la exportación, no contienen defectos o son lo suficientemente leves para no afectar su aspecto, calidad o conservación en general.
* ***Clase I:*** Destinados al consumo local, de buena calidad con algunos defectos como quemaduras por el sol, rozaduras, etc.
* ***Clase II:*** Reservados normalmente para el procesamiento industrial, cuentan con los mismos defectos de la Clase I a una escala mayor.

A picture containing fruit

Description automatically generated

Figura . Clases de mango disponible

# Diseño e implementación

El proceso de implementación de las técnicas de aprendizaje se basa en el propuesto en la guía de usuario de scikit-learn. Este se basa en un flujo de trabajo típico para este tipo de aplicaciones.

En primer lugar, se dividen los datos en dos grupos, uno de validación y otro de entrenamiento. Posteriormente, se entrenan y evalúan los modelos usando métodos de validación cruzada. Finalmente, se optimizan los hiperarametros de cada modelo para consecutivamente elegir el modelo mas adecuado. Se muestra un diagrama del flujo en el que se basa el proyecto:

Diagram

Description automatically generated

Figura . Flujo de trabajo típico

## Preprocesamiento de datos

Ya que uno de los modelos propuestos el CNN, cuyo objetivo es aprender características de imágenes, el único pre-procesado que se realizo sobre la base de datos fue el aumento de datos, utilizando la librería de tensor Flow, se generaron 1800 imágenes nuevas para cada clase.

A picture containing shape

Description automatically generated

Figura . Ejemplo de imágenes generadas das.

## Evaluación de modelos y optimización de hiperparametros

Las métricas utilizadas para la evaluación de todos los modelos serán, la exactitud, sensibilidad, precisión, f1-score y el área bajo la curva. Con estas métricas, se hará una evaluación mas completa de cada modelo mediante el método k-fold. Ya que el conjunto de datos es relativamente mediano (6000 imágenes), los datos de entrenamiento se dividirán en 5 grupos diferentes.

### Regresión logística (LR)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | test\_recall\_macro | test\_precision\_macro | test\_f1\_macro | test\_accuracy | test\_roc\_auc\_ovo |
| CV 1 | 0.95555556 | 0.95629895 | 0.95566859 | 0.955555556 | 0.987233153 |
| CV 2 | 0.9712963 | 0.97127897 | 0.9712566 | 0.971296296 | 0.987653035 |
| CV 3 | 0.96203704 | 0.96198103 | 0.96199796 | 0.962037037 | 0.984311343 |
| CV 4 | 0.96666667 | 0.96692443 | 0.96663911 | 0.966666667 | 0.993831662 |
| CV 5 | 0.95648148 | 0.95664422 | 0.9565369 | 0.956481481 | 0.986417181 |
| Average | 0.96240741 | 0.96262552 | 0.96241983 | 0.962407407 | 0.987889275 |

### K-Vecinos (KNN)

El clasificador KNN se encarga de almacenar los datos de entrenamiento y realiza la clasificación basándose en una votación de vecinos. Cada vez que se realiza una predicción, se le asigna una clase basado en la clase de los vecinos mas cercanos.

Uno de los algoritmos usados por el clasificador para escoger el vecino mas cercano es  *KD Tree*, que usa como uno de sus parámetros es el *Leaf size*  y será tomado en cuenta para el análisis de hiperparametros.

Otro parámetro a tener en cuenta es el numero de vecinos empleado para realizar la clasificación, tomando en cuenta la recomendación de que este numero sea menor a la raíz cuadrada del número de datos.

Finalmente se evalúa también el modelo con dos definiciones de distancia mínima diferente, la euclidiana y la de Minkowski. A continuación, se muestra u resumen de los parámetros utilizados en la optimización:

|  |  |
| --- | --- |
| Parámetros | Parámetros propuestos |
| Leaf size | [1, 11, 21, 31, 41] |
| K-vecinos | [1, 9, 17, 25, 33, 41] |
| Tipo de distancia[[1]](#footnote-2) | [Euclidiana, Minkowski] |

Tabla . Hiperparametros a optimizar.

#### Resultados:

Luego de evaluar las métricas propuestas usando validación cruzada se obtuvieron los siguientes resultados:

Chart, scatter chart

Description automatically generated

Figura . Resultados *Recall*

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

### Bosque aleatorio (RF)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **test\_recall\_macro** | **test\_precision\_macro** | **test\_f1\_macro** | **test\_accuracy** | **test\_roc\_auc\_ovo** |
| **CV 1** | 0.675926 | 0.72005 | 0.664506 | 0.675926 | 0.887653 |
| **CV 2** | 0.665741 | 0.703508 | 0.652739 | 0.665741 | 0.881247 |
| **CV 3** | 0.67963 | 0.724829 | 0.668267 | 0.67963 | 0.888029 |
| **CV 4** | 0.681481 | 0.730804 | 0.668852 | 0.681481 | 0.887236 |
| **CV 5** | 0.683333 | 0.72158 | 0.670704 | 0.683333 | 0.882288 |
| **Average** | 0.677222 | 0.720154 | 0.665014 | 0.677222 | 0.885291 |

### Redes neuronales convencionales (CNN)

## Comparación de modelos y selección

# Conclusiones

1. Representado en scikit-learn como p ( 1 = distancia euclídea, 2 = Minkowski ) [↑](#footnote-ref-2)