# Desarrollo de modelos basados en estructuras de Machine Learning que relacione parámetros de calidad del mango usando Imágenes hiperespectrales

**Carla V. Alcalde**
Universidad de Piura
Piura, Perú
carla.alcalde@alum.udep.edu.pe

**Vincenzo A. Luna**
Universidad de Piura
Piura, Perú
vincenzo.luna@alum.udep.edu.pe

**Alessandra Oquelis**
Universidad de Piura
Piura, Perú
alessandra.oquelis@alum.udep.edu.pe

**Rafael E. Pachas**
Universidad de Piura
Piura, Perú
rafael.pachas@alum.udep.edu.pe

**Paul C. Peña**
Universidad de Piura
Piura, Perú
paul.pena@alum.udep.edu.pe

***Resumen*** *- En este proyecto de investigación se estudiarán y evaluarán seis métodos para predecir los parámetros primarios del mango, que son los grados Brix y el estado de maduración, para crear un modelo de control de calidad automatizado, utilizando imágenes hiperespectrales y Machine Learning. El primer método consiste en predecir los parámetros utilizando el modelo de Partial Least Squares Regression (PLSR), aplicando K-Fold Cross-Validation. En el segundo y tercero, se utilizan los modelos Random Forest Regression (RFR) y Stochastic Gradient Boosting Regression (SGBR), respectivamente, para predecir el parámetro Brix. En el cuarto y quinto, se utilizan los modelos Random Forest Classification (RFC) y Stochastic Gradient Boosting Classification (SGBC), respectivamente, para predecir los parámetros aplicando un Label Encoder para obtener el estado de maduración del mango a partir de rangos numéricos establecidos. Finalmente, para el sexto método, los parámetros se predicen utilizando el modelo Support Vector Machine (SVM) aplicando los hiperparámetros C, gamma, grado y kernel para procesar los datos de forma óptima.*

***Keywords*** *- imágenes hiperespectrales, grados Brix, Partial Least-Squares Regression, K-fold Cross-Validation, Random Forest Regression, Stochastic Gradient Boosting Regression, Random Forest Classification, Stochastic Gradient Boosting Classification, Support Vector Machine.*

## 1. Introducción

Hoy en día, una de las frutas más comerciales y populares en todo el mundo es el mango. Esta fruta pertenece al tipo *Mangifera*, que posee una amplia variedad con 30 especies diferentes y su producción se expande a todos los territorios tropicales y subtropicales del mundo.

El mango tiene un alto valor comercial gracias a sus características, como el bajo contenido de calorías, riqueza en ácidos, presencia de vitaminas C, B5 y A. También es una de las mejores frutas antioxidantes que ayuda a la defensa contra la degradación celular en todo el cuerpo humano.

En Perú, 22 de los 24 departamentos producen mango apto para la exportación, destacando entre ellos el departamento de Piura como el mayor productor de todos, ya que produce el 66,7 % del total de mango cosechado. Según *AgrodataPerú*, en marzo de 2022, el Perú registró una ganancia total de 193 281 807 dólares en exportación y un promedio mensual de 64 427 269 dólares. Entre las especies cosechadas en el Perú destacan el Kent (el más cosechado y exportado), el Haden, el Edward y el Tommy Atkins.

A pesar de ser una de las industrias más grandes y sustentables de nuestro país, aún existen algunas dificultades para ejecutar el control de calidad mediante métodos no destructivos a grandes cantidades de mango.

Este proyecto de investigación presentará una serie de conceptos aplicados a: el análisis de imágenes hiperespectrales, el uso de Machine Learning y la evaluación de la calidad

del mango de acuerdo a sus parámetros fisicoquímicos; para posteriormente realizar el desarrollo de diferentes códigos utilizando los siguientes algoritmos: *Partial Least Squares (PLS), Support Vector Machine (SVM), Random Forest Regression (RFR) y Stochastic Gradient Boosting Regression (SGBR).*

## 2. Conceptos Generales

### 2.1. Factor de Inflación de Varianza

El Factor de Inflación de Varianza (VIF) se utiliza para medir la colinealidad entre las variables, lo que significa que determina el efecto que una variable causa en las demás. Al eliminar las que tienen valores de VIF superiores a 5, el número de variables se reducirá finalmente a las que se van a utilizar en el modelo, éstas se denominan componentes principales.

### 2.2. Partial Least-Squares Regression

Es un modelo de prendizaje supervisado que requiere etiquetas definidas y un objetivo para hacer predicciones sobre datos futuros. Reduce el número de variables combinando las que están altamente correlacionadas asignando pesos óptimos que dan una mayor varianza para una una mayor varianza para una mejor predicción.

*K-Fold Cross Validation:* Calcula el número óptimo de variables que deben utilizarse en el modelo mediante el cálculo del error cuadrático medio. Se elige la que tenga el mínimo error.

### 2.3. Support Vector Machine

*Support Vector Machine* transforma el conjunto de datos en una dimensión espacial relativa más grande, encontrando un clasificador de vectores de apoyo que puede clasificar los datos de forma eficaz.

### 2.4. Random Forest

*Random Forest* es la unión de varios modelos no robustos. Reduce el inconveniente del sobreajuste presente cuando se utilizan sólo ”árboles de decisión”, ya que los resultados de cada árbol se promedian. Normalmente, el número de características involucradas en cada árbol es la raíz del número de predictores.

Se crea un conjunto de datos *bootstrapped* con el mismo número de muestras que el conjunto de datos original, pero no necesariamente con las mismas muestras debido a que se permite repetir muestras.

Un cierto número de variables serán consideradas candidatas a ser el nodo raíz, la que tenga menos SSR será elegida y bloqueada. Algunas de las variables sobrantes se seleccionarán aleatoriamente y se utilizarán para construir el próximo nivel del árbol y así sucesivamente hasta terminar el árbol. Este procedimiento se repetirá con cada árbol.

Los datos que no se utilizaron en el *bootstrap* se denominan *Out-of-Bag dataset* y se utilizarán en la validación, ya que son datos que el modelo no ha utilizado todavía.

### 2.5. Stochastic Gradient Boosting

Se trata de un modelo ensamblado que utiliza árboles unidos en serie de forma que cada árbol se crea a partir del anterior, esto se hace para reducir los errores o pseudo-residuos.

Se crea una hoja que representa una suposición inicial de los pesos de las muestras. La media de los pesos supuestos de cada muestra es el peso de la hoja.

Se creará una nueva columna con los pseudo-residuos de cada muestra.

## 3. Metodología

### 3.1. Recolección de la data

Para este proyecto se seleccionó un grupo de 85 mangos Kent para analizar los aspectos importantes a evaluar en un control de calidad previo a la exportación: la dulzura y madurez de la fruta.

#### 3.1.1. Imágenes hiperespectrales

La madurez se analizó mediante el uso de la teledetección, concretamente la tecnología de imágenes hiperespectrales. Para cada mango se tomó una foto con una cámara especial (Modelo Pika, II generación) junto con el software *Resonon*.

Por cada foto, se seleccionaron 4 regiones de interés *(ROIs)* diferentes, y cada una de ellas se obtuvo una firma espectral respectiva, la cual es un gráfico que muestra los distintos valores de reflectancia que tiene el mango a lo largo de 240 bandas.

#### 3.1.2. Grados Brix

Después de la recolección de imágenes se midieron los grados Brix con el sensor DMA 35, que cuenta con diferentes funciones de medición. Sin embargo, para los fines de esta investigación se midió el nivel de azúcar en el jugo de mango.

Para esto, cada mango fue cortado y licuado para obtener el jugo, el cual fue introducido en el sensor, que después de estabilizarse, proporcionó la información necesaria.

#### 3.1.3. Creación del dataset

Se obtuvo un total de 341 firmas hiperespectrales, es decir, 341 valores de datos de las 240 bandas de reflectancia.

De los datos originales, el 15 % se separó al azar para crear el conjunto de datos de prueba que se utilizarán para comprobar los modelos. Y el 85 % restante se utilizó para entrenar y validar los modelos.

### 3.2. Creación de los Modelos

Se crearon 4 modelos de regresión: *Support Vector Machine, Random Forest Regression, Stochastic Gradient Regression, Partial Least Squares*, y 2 modelos de clasificación: *Random Forest Classification y Stochastic Gradient Classification*. En cada uno de los modelos, los valores objetivo

están representados por la matriz **"Y"**, mientras que los valores de las características están representados por la matriz **"X"**. Además, en cada modelo se importan las librerías necesarias de los distintos paquetes que proporciona Python y el archivo Excel que se utilizará para el modelo desarrollado.

Para los modelos de regresión, los datos se ordenan en 2 matrices: una que contiene los valores objetivo (Brix), mientras que la otra contiene los valores de las características (Bandas). De las características, se omite la columna de "Brix", ya que el valor de Brix es la predicción. Después, ambas matrices se convierten en *data-frames*.

#### 3.2.1. Partial Least Squares Regression

Una vez definidas las variables implicadas, se ejecuta un preprocesamiento de los datos conocido como normalización, que consiste en un escalado y una normalización. A continuación, se analiza la matriz **"Y"** y se encuentra un error en sus dimensiones, por lo que se aplica una remodelación de la matriz para obtener la forma correcta para la aplicación del modelo.

Tras la remodelación, los datos se procesan para ejecutar el entrenamiento y la prueba del modelo. En este caso, el 40 % de los datos se estableció para la prueba y el otro 60 % para el entrenamiento. Además, se utiliza una validación cruzada de K-fold para probar lo bien que predice el modelo, utilizando datos que no se han utilizado para la construcción del modelo.

Posteriormente, se calcularon los errores medios al cuadrado en un rango de 1 a 20. Según este gráfico, el modelo PLSR mejorará si el valor de los componentes "n.[es] igual a 10.

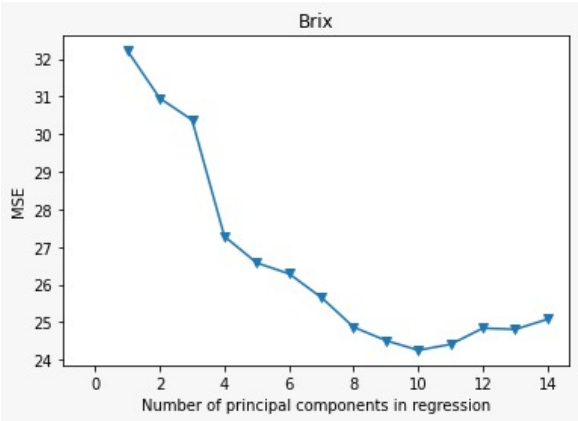


Figura 1. Gráfico de número de componentes principales vs mse

Los valores resultantes de las métricas $R2_{score-train}$ y $R2_{score-test}$ tras la creación del modelo son muy diferentes entre sí. Son iguales a 0.330638 y 0.228501 respectivamente.

Esto muestra que el modelo está subajustado, porque $R2_{score-train}$ y $R2_{score-test}$ son inferiores a 0.5, por lo que se puede anticipar que este modelo predecirá mal los resultados. z

#### 3.2.2. Support Vector Machine

Una vez definidas las variables implicadas, se ejecuta un preprocesamiento de los datos conocido como escalado. Después, se analiza el *array* **"Y"** y se encuentra un error en sus dimensiones, por lo que se aplica una remodelación del *array* para obtener la forma correcta para la aplicación del modelo.

Tras la remodelación, los datos se procesan para ejecutar el entrenamiento y la prueba del modelo. En este caso, el 30 % de los datos se estableció para la prueba y el otro 70 % para el entrenamiento, utilizando un valor de estado aleatorio igual a 100.

Estableciendo los hiperparámetros del modelo, se seleccionó un kernel polinómico, un valor de grado de 2 con un valor de gamma de 0.1, y un C igual a 10.

Los valores resultantes de las métricas $Accuracy_{score-train}$ y $Accuracy_{score-test}$ tras la creación del modelo obtenido no se parecen entre sí. Para el modelo SVMR, las métricas obtenidas son iguales a 0.212632 y 0.292334 respectivamente.

Analizando estos valores, se puede afirmar que el modelo está subajustado, debido a que $Accuracy_{score-train}$ y $Accuracy_{score-test}$ son inferiores a 0.5, por lo que se puede concluir que este modelo predecirá resultados muy pobres.

#### 3.2.3. Random Forest Regression y Stochastic Gradient Boosting Regression

Para la elaboración de RFR y SGBR, como en los modelos anteriores, los datos se ordenan en 2 matrices: una que contiene los valores objetivo (Brix), mientras que la otra contiene los valores de las características (Bandas). De las características, se omite la columna de "Brix", ya que el valor de Brix es la predicción. Después, ambas matrices se convierten en *data-frames*.

Después, se analiza el *array* **"Y"** y se encuentra un error en sus dimensiones, por lo que se aplica una remodelación del *array* para obtener la forma correcta para la aplicación del modelo.

En este caso, se estableció un 20 % de los datos para la prueba y el otro 80 % para el entrenamiento, utilizando un valor de estado aleatorio igual a 100.

Los valores resultantes de las métricas $R2_{score-train}$ y $R2_{score-test}$ tras la creación de los modelos con todos los datos, difieren. Para el modelo RFR, las métricas obtenidas fueron 0.856988 y 0.160498 respectivamente, mientras que para el modelo SGBR fueron 0.905964 y 0.102999.

Analizando estos valores, se puede afirmar que los modelos están sobreajustados, debido a que los valores de $R2_{score-train}$ son muy cercanos a 1 y los de $R2_{score-test}$, a 0. Por lo tanto, se puede anticipar que este modelo no dé buenos resultados.

#### 3.2.4. Random Forest Classification y Stochastic Gradient Boosting Classification

Para los modelos de clasificación se añade una columna que indica si el mango está maduro o no. Inmediatamente los datos se clasifican en 2 matrices: una que contiene los

valores objetivo (maduro o no maduro), mientras que la otra contiene los valores de las características (Bandas). Después, ambas matrices se convierten en *data-frames*

Una vez definidas las variables implicadas, se ejecuta en las características un preprocesamiento de datos conocido como escalado, que transforma los valores en un rango de 0 a 1. A continuación, se analiza el *array* **"Y"** y se encuentra un error en sus dimensiones, por lo que se aplica una remodelación del *array* para obtener la forma correcta para la aplicación del modelo.

Se ejecuta otro preprocesamiento, al objetivo esta vez, conocido como *Label Encoder*, que transforma el objetivo en valores de 0 y 1.

En este caso, el 20 % de los datos se estableció para la prueba y el otro 80 % para el entrenamiento, utilizando un valor de estado aleatorio igual a 80.

Los valores resultantes de las métricas $Accuracy_{score-train}$ y $Accuracy_{score-test}$ tras la creación de los modelos con todos los datos obtenidos no se parecen entre sí. Para el modelo RFC, las métricas obtenidas son iguales a 1 y 0.840580 respectivamente, mientras que para el modelo SGBC fueron iguales a 1 y 0828087. Estos valores muestran que los modelos están ajustados, ya que el $Accuracy_{score-train}$ es igual a 1 y el $Accuracy_{score-test}$ es muy cercano, por lo que se puede anticipar que estos modelos darán una buena clasificación.

## 4. Desarrollo y Experimentación

Para los modelos que se presentarán a continuación, los datos utilizados son los mismos que figuran en la sección de metodología. Cabe señalar que para todos los modelos los datos mencionados se han dividido al azar en un 85 % para la construcción y el 15 % restante para la validación mediante la función *test(datatest)*.

Con el objetivo de encontrar los valores predichos, compararlos con los reales y encontrar su porcentaje de error relativo.

### 4.1. Partial Least Squares Regression

Con las métricas obtenidas del modelo *Partial Least Squares Regression* (PLS), se concluye que el modelo está subajustado, es decir, que no entrena ni prueba bien, por lo que el modelo predecirá mal.

**Resultados** A continuación se muestra la comparación entre los valores reales y los valores predichos por el modelo PLS.

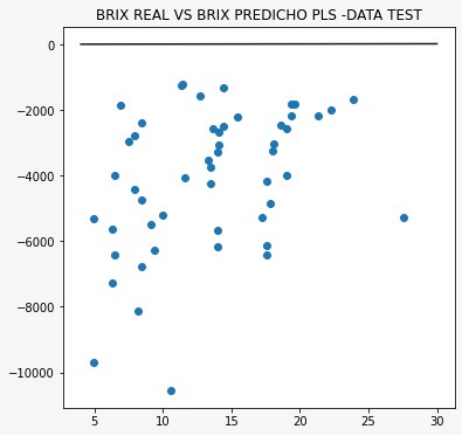


Figura 2. °Brix real vs. °Brix predicho por PLS

### 4.2. Support Vector Machine

Con las métricas obtenidas del modelo *Support Vector Machine* (SVM), se concluye que el modelo está subajustado, es decir, que no entrena ni prueba bien, por lo que el modelo predecirá mal.

**Resultados** A continuación se muestra la comparación entre los valores reales y los valores predichos por el modelo SVM.

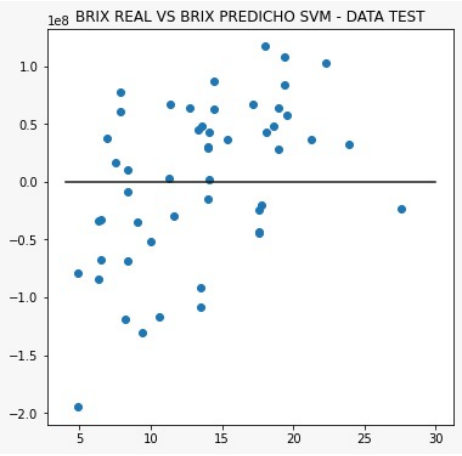


Figura 3. °Brix real vs. °Brix predicho por SVM

### 4.3. Random Forest Regression y Stochastic Gradient Boosting Regression

Con las métricas obtenidas de ambos modelos de regresión, se concluye que ambos modelos están sobreajustados, por lo que el modelo predecirá mal si se introducen nuevos datos.

**Resultados** A continuación se muestra la comparación entre los valores reales y los valores previstos por los modelos RFR y SGBR.

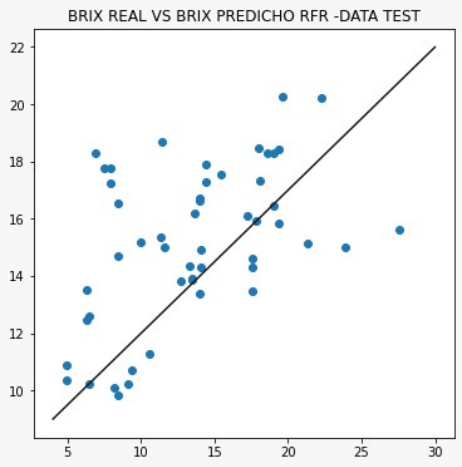


Figura 4. °Brix real vs. °Brix predicho por RFR

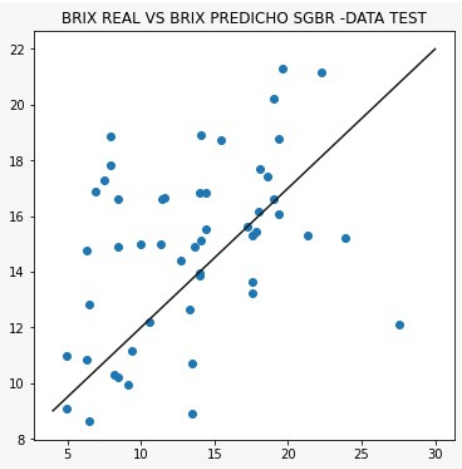


Figura 5. °Brix real vs. °Brix predicho por SGBR

### 4.4. Random Forest Classification y Stochastic Gradient Boosting Classification

Con las métricas obtenidas de ambos modelos de clasificación, se concluye que ambos modelos se ajustan, por lo que el modelo predecirá bien si se introducen nuevos datos.

**Resultados** A continuación se muestra la comparación entre los valores reales y los valores previstos por los modelos RFC y SGBC.

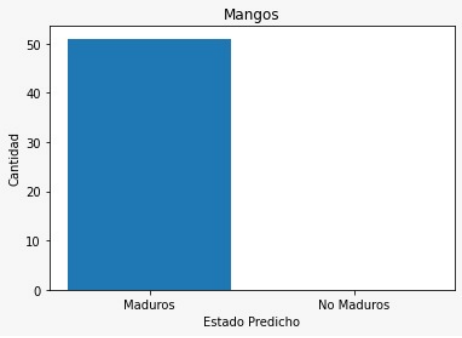


Figura 6. °Brix real vs. °Brix predicho por RFC

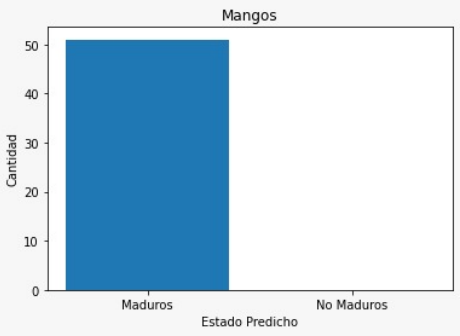


Figura 7. °Brix real vs. °Brix predicho por SGBC

## 5. Análisis y Discusión de los Resultados

Como ya se ha dicho, los datos se probaron dos veces. Primero con todos los datos obtenidos y luego con el 85 % debido a la necesidad de validar los modelos con nuevos datos.

### 5.1. Partial Least Squares Regression

Los valores resultantes de las métricas $R2_{score-train}$ y $R2_{score-test}$ tras la creación del modelo con el 85 % de los datos, son muy diferentes entre sí. Son iguales a 0.326024 y 0.18242435 respectivamente, se mantuvo el subajuste.

El modelo tiene un error relativo máximo igual a 198221.198 % y un error relativo mínimo igual a 7031.519 %, lo que confirma que este modelo predice los grados Brix de manera incorrecta.

### 5.2. Support Vector Machine

Los valores resultantes de las métricas $Accuracy_{score-train}$ y $Accuracy_{score-test}$ tras la creación del modelo con el 85 % de los datos son muy diferentes entre sí. Siendo iguales a 0.005743 y 0.282306, respectivamente, manteniendo el subajuste.

El modelo arroja valores predichos muy alejados de los reales. Presenta un error relativo máximo igual a -2649810878 % y un error relativo mínimo igual a -53423978 % confirmando que este modelo predice mal los grados Brix.

### 5.3. Random Forest Regression y Stochastic Gradient Boosting Regression

Los valores resultantes de las métricas $R2_{score-train}$ y $R2_{score-test}$ tras la creación de los modelos con el 85 % de los datos varían mucho entre sí.

Para el modelo RFR las métricas obtenidas son iguales a 0.856933 y 0.128938 respectivamente, mientras que para el modelo SGBR son iguales a 0.928735 y -0.002783, manteniendo el sobreajuste.

Ambos modelos arrojan valores predichos que no son precisos. El modelo RFR presenta un error relativo máximo igual a -165 % y un error relativo mínimo igual a -165 %. Mientras que el modelo SGBR presenta un error relativo

máximo igual a -144 % y un error relativo mínimo igual a -1 % y un error relativo máximo igual a -144 % y un error relativo mínimo igual a 1 %, confirmando así que ambos modelos predecirán los grados Brix erróneamente.

### 5.4. Random Forest Classification y Stochastic Gradient Boosting Classification

Los valores resultantes de las métricas $Accuracy_{score-train}$ y $Accuracy_{score-test}$ tras la creación de los modelos con el 85 % de los datos son cercanos en ambos modelos.

Para el modelo RFC las métricas obtenidas son iguales a 1 y 0.793103 respectivamente, mientras que para el modelo SGBC son iguales a 1 y 0.775862, manteniendo el ajuste de los modelos.

Ambos modelos muestran que el total de los datos utilizados para la validación representan mangos maduros, no coincidiendo en el 100 %, sino sólo en el 72 %, ya que no logra clasificar 14 ROIs.

Tabla 1. Métricas obtenidas por los modelos con el 85 % de la data

| | Grados Brix | | | |
|---|---|---|---|---|
| Métricas | PLS | SVM | RFR | SGBR |
| $R2_{score-train}$ | 0.326024 | - | 0.856933 | 0.928735 |
| $R2_{score-test}$ | 0.182435 | - | 0.128938 | -0.002783 |
| $Accuracy_{score-train}$ | - | 0.005743 | - | - |
| $Accuracy_{score-test}$ | - | 0.282306 | - | - |
| | Estado del mango | | | |
| Métricas | RFC | | SGBC | |
| $Accuracy_{score-train}$ | 1 | | 1 | |
| $Accuracy_{score-test}$ | 0.793103 | | 0.77586 | |

## 6. Conclusiones

En la experimentación del modelo *Partial Least Squares Regression*, se obtuvo una precisión del 18.24 % con la data del testeo, mientras que con la data del entrenamiento se obtuvo una precisión del 32.6 %. Debido a estos valores bajos, se optó por aplicar VIF a nuestro modelo. Sin embargo, no se consiguió una mejora, por lo cual se descartó su aplicación.

El modelo de *Support Vector Machine* no muestra la mejor precisión frente a la regresión porque a pesar de que se ha implementado el hiperparámetro C para controlar el error, el kernel polinomial y radial no brindan los mejores resultados al utilizarlos para el método de Machine Learning. Esto se debe a que el modelo de SVM es aplicable para un dataset con baja colinealidad entre sus features y al analizar la base de datos haciendo uso del VIF se puede ver que este no es el caso, por lo tanto, este modelo de regresión no es el adecuado para el dataset que se tiene.

En los modelos de *Random Forest Regression* y *Stochastic Gradient Boosting Regression* hubo resultados de precisión: para el primer modelo mencionado del 85.7 % con la data del entrenamiento y el 12.89 % con la data del testeo; y para el segundo modelo mencionado unos valores de precisión del 92.87 % para la data del entrenamiento y -0.278 % para la data del testeo. Esto nos demuestra que los modelos presentan un valor bajo de sesgo, lo que significa que la data de entrenamiento se ajusta a la data real, además presentan un valor alto de varianza, lo cual nos indica que los modelos se ajustan a la data de entrenamiento mas no a la data de testeo.

En los modelos de *Random Forest Classification* y *Stochastic Gradient Boosting Classification* hubo resultados de precisión: para el primer modelo mencionado del 100 % con la data del entrenamiento y 79.31 % con la data del testeo; y para el segundo modelo mencionado unos valores de precisión del 100 % para la data del entrenamiento y 77.59 % para la data del testeo. Esto nos demuestra que los modelos presentan un valor bajo de sesgo, lo que significa que la data de entrenamiento se ajusta a la data real, además presentan un valor bajo de varianza, lo cual nos indica que los modelos se ajustan tanto a la data de entrenamiento como a la data de testeo. Por lo tanto, se concluye que *Random Forest Classifcation* es el mejor modelo para los fines de esta investigación.

Luego de experimentar con los modelos elaborados, se concluye que los datos obtenidos sobre los grados Brix de los mangos no guardan una relación coherente con los datos de las bandas tomadas, lo cual puede haber sido ocasionado por el sensor utilizado durante la medición del parámetro. Esta incoherencia entre los datos se ve reflejada posteriormente en las métricas halladas en cada modelo.

Los modelos de clasificación nos dan una mejor precisión que los modelos de regresión debido a que en la clasificación se toman dos rangos de grados Brix y se etiquetan creando así dos clases en el target, mientras que en la regresión se trata de predecir exactamente los grados Brix siendo algo mucho más complejo, teniendo en cuenta lo mencionado en la primera conclusión.


## Agradecimientos

Nos gustaría agradecer a nuestro asesor MsC. Juan Carlos Soto Bohórquez por su tiempo y apoyo a lo largo de este proyecto, así como al Dr. William Ipanaqué Alama por guiarnos a lo largo del curso de SAC. Al Laboratorio de Sistemas Automáticos de Control por proporcionarnos las herramientas necesarias cuando se requirió. Finalmente, también un agradecimiento especial a nuestras familias por su tiempo y apoyo incondicional durante la ejecución de este proyecto.


## Referencias


[1] NANDI, C. S., TUDU, B., & KOLEY, C. (2016). *A Machine Vision Technique for Grading of Harvested Mangoes Based on Maturity and Quality.* IEEE Sensors Journal, 16(16), 6387–6396. https://doi.org/10.1109/jsen.2016.2580221

[2] ZHANG, B., HUANG, W., LI, J., ZHAO, C., FAN, S., WU, J., & LIU, C. (2014). *Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review.* Usda.gov. https://pubag.nal.usda.gov/catalog/5432170

[3] ABBOTT, J. A. (1999). *Quality measurement of fruits and vegetables*. Postharvest Biology and Technology, 15(3), 207–225. https://doi.org/10.1016/s0925-5214(98)00086-6

[4] GOETZ, A. F. H., VANE, G., SOLOMON, J. E., & ROCK, B. N. (1985). *Imaging Spectrometry for Earth Remote Sensing*. Science, 228(4704), 1147–1153. https://doi.org/10.1126/science.228.4704.1147

[5] TREVOR A. CRANEY & JAMES G. SURLES (2002) Model-Dependent Variance Inflation Factor Cutoff Values, Quality Engineering, 14:3, 391-403, DOI: https://doi.org/10.1081/QEN-120001878

[6] RUNGPICHAYAPICHET, P., NAGLE, M., YUWANBUN, P., KHUWIJITJARU, P., MAHAYOTHEE, B., & MÜLLER, J. (2017). *Prediction mapping of physicochemical properties in mango by hyperspectral imaging*. Biosystems Engineering, 159, 109–120. https://doi.org/10.1016/j.biosystemseng.2017.04.006

[7] CASTRO SILUPU, W. M. (2022). *Aplicación de la tecnología de imágenes hiperespectrales al control de calidad de productos agroalimentarios de la región de Amazonas (Perú)*. https://doi.org/10.4995/thesis/10251/63250

[8] BRENES ZELEDÓN, C. (2019). *Captura y formación de imágenes hiperespectrales mediante UAV's*. Tecnología En Marcha, 32(6), 24–34. https://dialnet.unirioja.es/servlet/articulo?codigo=7450239

[9] VIERA-MAZA, G. (2018). *Aplicación de procesamiento de imágenes para clasificación de granos de cacao según su color interno*. April 10, 2022, from https://pirhua.udep.edu.pe/bitstream/handle/11042/3486/MAS_IME_AUT_030.pdf?sequence=2&isAllowed=y

[10] GARETH JAMES, DANIELA WITTEN, TREVOR HASTIE & ROBERT TIBSHIRANI. (2013). *An Introduction to Statistical Learning*. New York: Springer Science+Business Media.

[11] GUIDO, ANDREAS C. MÜLLER & SARAH. (2017). *Introduction to Machine Learning with Python.* Sebastopol: O'Reilly.

[12] P. GELADI & B. R. KOWALSKI, *"Partial least-squares regression: a tutorial"*, Analytica Chimica Acta, vol. 185, pp. 1–17, 1986, doi:10.1016/0003-2670(86)80028-9.

[13] BREIMAN, L., BERKELEY, U., & CUTLER, A. (n.d.). *Random Forests for Scientific Discovery*. https://www.usu.edu/math/adele/RandomForests/ENAR.pdf

[14] FREUND, Y., & SCHAPIRE, R. E. (1997). *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. Journal of Computer and System Sciences, 55(1), 119–139. https://doi.org/10.1006/jcss.1997.1504

[15] JEROME H. FRIEDMAN. (2002). *Stochastic gradient boosting*. Computational Statistics and Data Analysis, 38(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

[16] URTUBIA, A., LEÓN, R., & VARGAS, M. (2021). *Identification of chemical markers to detect abnormal wine fermentation using support vector machines*. Computers and Chemical Engineering, 145, 107158. https://doi.org/10.1016/j.compchemeng.2020.107158