FACULTAD DE INGENIERIA GEOGRAFICA AMBIENTAL Y ECOTURISMO

“ESTIMACIÓN ESPACIAL DE LA SUSCEPTIBILIDAD DE MOVIMIENTOS EN MASA MEDIANTE APRENDIZAJE AUTOMÁTICO EN LA SUB CUENCA CHECRAS”

Línea de investigación:
Procesamiento digital de imágenes y señales

Informe de suficiencia profesional para optar el Título Profesional de Ingeniero Geógrafo

Autor:
Hansen Wibelsman Bueno Gómez

Asesor:
Marco Antonio Herrera Díaz
ORCID: 0000-0002-8578-4259

Jurado:
Altez Rodriguez, Jose Felix
Aguirre Cordero, Rogelio
Paricoto Simon, Maria Mercedes

Lima – Perú
2023

“ESTIMACIÓN ESPACIAL DE LA SUSCEPTIBILIDAD DE MOVIMIENTOS EN MASA MEDIANTE APRENDIZAJE AUTOMÁTICO EN LA SUB CUENCA CHECRAS”

INFORME DE ORIGINALIDAD

18%	17%	9%	6%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	hdl.handle.net Fuente de Internet	2%
2	cybertesis.uni.edu.pe Fuente de Internet	1%
3	repositorio.ingemmet.gob.pe Fuente de Internet	1%
4	repositorio.unfv.edu.pe Fuente de Internet	1%
5	www.researchgate.net Fuente de Internet	1%
6	Submitted to Sheffield Hallam University Trabajo del estudiante	1%
7	alicia.concytec.gob.pe Fuente de Internet	<1%
8	link.springer.com Fuente de Internet	<1%

Dedicatoria

Este trabajo está dedicado a mis padres y hermanos, quienes constantemente me brindan su apoyo e impulsan a ser mejor cada día y en todo momento mediante, sus acciones y palabras que fortalecen todas mis ganas por continuar aprendiendo.

A mis amistades, quienes aportan personal y académicamente en mí, compartiendo conocimientos y experiencias invaluables.

A mis profesores y mentores, cuya guía y enseñanza han sido fundamentales en mi crecimiento académico, abriéndome puertas a muchas posibilidades de desarrollo profesional.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE DIAGRAMAS

ÍNDICE DE ECUACIONES

ÍNDICE DE FIGURAS

Resumen

El propósito fundamental de este informe radica en destacar la importancia de la investigación en todos los campos relacionados con el territorio y el medio ambiente, y cómo puede servir como recurso para generar soluciones concretas ante un desafío que ejerce un impacto constante en nuestra sociedad, como lo son los movimientos en masa. A través de la aplicación de mi experiencia laboral adquirida en proyectos relacionados con la implementación y desarrollo de la teledetección y la programación, este informe aspira a proporcionar un enfoque más sofisticado para abordar este problema. En esencia, el objetivo general que guía este esfuerzo es la estimación espacialmente la susceptibilidad de movimientos en masa mediante aprendizaje automático en la subcuenca Checras. La investigación se clasificó como aplicada, ya que está orientada a generar conocimiento mediante aplicaciones prácticas y a la resolución de problemas. Dado que se involucran datos cualitativos y cuantitativos, el diseño es de enfoque mixto. El nivel de investigación es correlacional, ya que se estudia la relación entre dos o más variables y también tiene un componente predictivo. Se aplicaron los modelos de máquina de soporte vectorial (SVM) y bosques aleatorios (RF), cada uno con dos métodos de preprocesamiento, y se obtuvieron resultados de evaluación con una curva ROC de 0.88 y 0.899 para SVM, y 0.900 y 0.908 para RF en el primer y segundo método, respectivamente. Esto demuestra que el modelo RF presenta un mejor rendimiento en comparación con SVM.

Palabras clave: aprendizaje supervisado, movimientos en masa, susceptibilidad

Abstract

The primary purpose of this report is to underscore the significance of research in all fields related to territory and the environment and how it can serve as a resource for generating concrete solutions to a challenge that continually impacts our society, such as mass movements. By drawing upon my work experience acquired in projects related to remote sensing and programming implementation and development, this report aims to provide a more sophisticated approach to addressing this issue. Essentially, the overarching goal guiding this effort is the spatial estimation of susceptibility to mass movements through machine learning in the Checras sub-basin. The research was categorized as applied, as it is oriented towards generating knowledge through practical applications and problem-solving. Given the involvement of qualitative and quantitative data, the design follows a mixed-methods approach. The research level is correlational, as it explores the relationship between two or more variables and also incorporates a predictive component. Machine learning models, specifically Support Vector Machines (SVM) and Random Forests (RF), were employed, each with two preprocessing methods. Evaluation results yielded ROC curves of 0.88 and 0.899 for SVM and 0.900 and 0.908 for RF in the first and second methods, respectively. This demonstrates that the RF model outperforms SVM.

Keywords: mass movements, supervised learning, susceptibility

I. INTRODUCCIÓN

En los últimos años, la inteligencia artificial ha ocasionado muchos cambios en la investigación de diversas disciplinas en el mundo entero. La aplicación de esta tecnología al campo de la geociencia no ha sido la excepción debido a que tiene diversas aplicaciones en esta disciplina como la determinación de posibles zonas de movimientos en masa, inundaciones, etc. Los movimientos en masa representan amenazas para comunidades y ecosistemas. Habitualmente, los estudios de determinación de susceptibilidad a movimientos en masa se desarrollaban en función a métodos determinísticos que se expresa en el factor de seguridad y heurísticos en el que se asignan pesos a los factores, pero el empleo de la inteligencia artificial ha dado una nueva perspectiva en predicción y precisión a estos estudios.

El presente informe está conformado por siete capítulos. El capítulo uno se presenta un breve resumen de la trayectoria laboral del autor, descripción de su centro laboral y las labores que desarrolla en ella. El capítulo dos y tres están compuestos por la actividad desarrollada en la que es realizado en la determinación de la susceptibilidad a movimientos en masa mediante la aplicación de aprendizaje supervisado y los resultados obtenidos. En el capítulo cuatro y cinco se desarrolla las conclusiones y las recomendaciones a partir de los resultados obtenidos. En el capítulo seis y siete se presentan las referencias que se usaron para el desarrollo del trabajo y los anexos obtenidos.

1.1. Trayectoria del Autor

El autor es bachiller en la carrera de Ingeniería Geográfica en la facultad de Ingeniería Geográfica, Ambiental y Ecoturismo (FIGAE) de la Universidad Nacional Federico Villarreal (UNFV) y continuó su capacitación en numerosos campos como la teledetección, fotogrametría, Sistemas de Información Geográfica (SIG), Machine Learning (ML), etc.

En el mundo laboral se inició en el rubro de la fotogrametría en el Instituto Geográfico Nacional (IGN), en el cual se desempeñó, inicialmente, como practicante en fotogrametría y posteriormente como fotogrametrista participando en la restitución de mapas catastrales de diversos proyectos a diversas escalas.

Posteriormente, en el Instituto Nacional de Estadística e Informática (INEI) se desempeñó como asistente automatizador y automatizador cartográfico de los censos del 2017.

En el Instituto Geológico, Minero y Metalúrgico (INGEMMET), se desempeñó como practicante profesional en el área de Laboratorio de Teledetección de la dirección de Laboratorios en el que realizó diversos servicios mediante la utilización de productos radar y ópticos.

En el Organismo de Formalización de la Propiedad Informal (COFOPRI) se desempeñó como fotogrametrista y como especialista en fotogrametría mediante la captura y utilización de productos de drones.

Más adelante, en la empresa Geocenter Ingenieros S.A.C. participó como operador fotogramétrico en la elaboración de productos fotogramétricos.

En la consultora INGEPRASEG Consultores S.A.C. desempeñándose como analista cartográfico en la elaboración de diversos mapas.

Actualmente, en el Organismo de Evaluación y Fiscalización Ambiental (OEFA), labora como especialista en sistematización de resultados de la información fotogramétrica de diversos sub sectores.

1.2. Descripción de la Institución

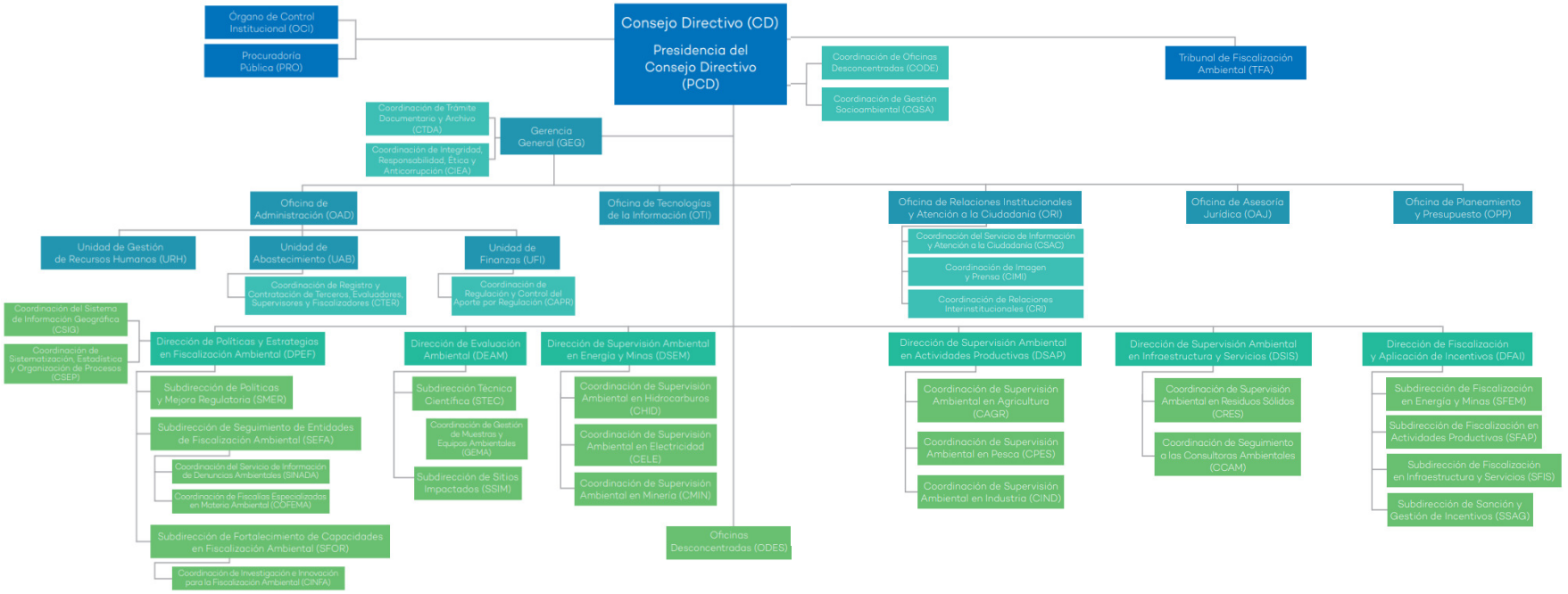
Organismo de Evaluación y Fiscalización Ambiental (OEFA), es un organismo público adscrito al Ministerio del Ambiente el cual promueve el cumplimiento de las responsabilidades ambientales con el fin de mantener un equilibrio entre las actividades económica y la protección ambiental.

Entre sus funciones principales de fiscalización directa se encuentra la de evaluadora para prevenir impactos ambientales y precisar responsabilidades mediante acciones de vigilancia, monitoreo y similares; supervisión directa con la que aseguren el cumplimiento de las obligaciones de las empresas mediante acciones de seguimiento, verificación y medidas administrativas; fiscalización y sanción en la que investigan y sancionan probables infracciones, además de aplicar sanciones por incumplimientos de los instrumentos de gestión ambiental, normas ambientales, compromisos ambientales y disposiciones emitidos por el OEFA.

1.3. Estructura orgánica

Figura 1.

Estructura orgánica del Organismo de Evaluación y Fiscalización Ambiental – OEFA, 2019



Nota. Tomado de Fiscalización ambiental: Construyendo confianza y facilitando la inversión (p. 10 y 11), por OEFA, 2019

1.4. Áreas y funciones desempeñadas

En el OEFA me desempeño como especialista en sistematización de resultados de la información geoespacial fotogramétrica de diversos proyectos enmarcados en sub sectores en el la Coordinación de Sistemas de Información Geográfica (CSIG), en la que mediante diversos softwares como ArcGIS Pro y lenguajes de programación como Python se realizan las siguientes actividades:

- Desarrollo de modelos de geoprocesamiento en model builder para el procesamiento de datos tabulares, vectoriales y ráster.
- Desarrollo de algoritmos en Python para el tratamiento y control de calidad de imágenes, datos tabulares y vectoriales.
- Procesamiento de imágenes de drones de modelos phantom 4 y mavic 2 enterprise dual del modelo DJI mediante softwares como Drone2Map.

II. DESCRIPCIÓN DE UNA ACTIVIDAD ESPECÍFICA

La estimación espacial de la susceptibilidad se realizó mediante la aplicación de técnicas computacionales avanzadas para comprender, evaluar y predecir áreas propensas a sufrir deslizamientos de tierra, avalanchas y otros fenómenos geológicos de movimientos en masa. Al analizar los factores condicionantes como datos topográficos, geológicos, etc. se busca patrones con las que se pueden identificar zonas con un alto riesgo. Los modelos predictivos permiten la toma de decisiones informadas para realizar una buena gestión territorial y ambiental.

Para el desarrollo del presente trabajo, se consideraron dos metodologías de trabajo en las que se usaron dos tipos de modelos de aprendizaje automático, con los que se obtuvieron los productos esperados.

2.1. Antecedentes

El estudio en la geociencia mediante el uso aprendizaje automático es relativamente nuevo en el mundo como los modelos máquina de soporte vectorial y bosques aleatorios. Esta ciencia se ha aplicado en la predicción e identificación de zonas susceptibles a distintos tipos de desastres por fenómenos naturales mediante el uso de los sistemas de información Geográfica (GIS), teledetección y otras técnicas.

2.1.1. Nacionales

Fídel y Zavala (2006) realizan el estudio de la susceptibilidad a los movimientos en masa de la cuenca de la quebrada Hualanga que se ubica en el departamento de La Libertad. En dicho estudio los autores utilizan el método multivariado evaluación multicriterio de las jerarquías analíticas con algunas modificaciones en la que toman en consideración de los pesos de los factores condicionantes utilizados.

Vilchez y Medina (2008) desarrollan un estudio en las áreas de Chachapoyas y Luya, departamento de Amazonas, enfocado a la susceptibilidad a los movimientos en masa mediante la aplicación de un método estadístico bivariante, con la cual determinaron los pesos que los factores condiconantes adquieren para la ocurrencia de movimientos en masa. Estos valores son reclasificados en términos de susceptibilidad y el mapa final es determinado mediante una adición entre las capas de cada uno de los factores. En este método los autores consideran como factores condicionantes a la litología, la geomorfología, la cobertura vegetal y la pendiente de laderas.

El territorio peruano ya cuenta con un mapa de susceptibilidad por movimientos en masa a escala 1:1 000 000. Villacorta, Fidel y Zavala (2012), explican el mapa mencionado fue elaborado con el método heurístico multivariado y para la validación usaron el mapa de peligros geológicos inventariado en el año 2000 a 2009 por la Dirección de Geología Ambiental y Riesgo Geológico (DGAR) del (INGEMMET). Los autores concluyen que como resultado las zonas de mayor susceptibilidad a movimientos en masa son:

1. Al oeste, entre los departamentos de Cajamarca, La Libertad, Ancash, Lima y Huancavelica.
2. Suroriental, Ayacucho, Apurímac, Cusco, Puno.
3. Suroccidental, Arequipa, Moquegua, Tacna.
4. Central y Nororiental, Junín, Pasco, Huánuco, San Martín.

2.1.2. Internacionales

Chang et al. (2020) utilizaron modelos de aprendizaje automático para realizar predicciones de la susceptibilidad de deslizamientos en la provincia de Jianxi en China en la cual encontraron 446 deslizamientos. Los autores aplicaron dos técnicas, aprendizaje automático supervisado con modelos como Máquina de soporte vectorial (SVM) y CHI-cuadrado Detección de Interacción Automática (CHAID) y aprendizaje automático no

supervisado con los modelos K-Means y Kohonen, las cuales son aplicados a 12 factores condicionantes que fueron obtenidos a través del procesamiento de imágenes satelitales Landsat 8, imágenes aéreas de alta resolución, análisis espacial de la topografía e hidrología y un Modelo de Digital de Elevación. El rendimiento de estos modelos fue evaluado mediante el Área Bajo la Curva (AUC) de la Curva de Característica Operativa del Receptor (ROC) con resultados de AUC de 0.892 para SVM y 0.872 para CHAID. Además, se usó Proporción de Frecuencia (FR) dando como resultado una precisión de 77.80% para SVM, seguido de CHAID con 74.50%, K-Means de 69.7% concluyendo que el SVM tiene una mejor precisión.

El estudio realizado por Arabameri et al. (2020) evaluaron la susceptibilidad de deslizamientos en la cuenca del río Galicash al norte de Irán, en la cual consideró dieciséis factores entre condicionantes y desencadenantes, y los deslizamientos fueron obtenidos mediante el uso de imágenes de alta resolución, mapas topográficos y puntos GPS obtenidos en el área de estudio. Los autores usaron tres modelos de aprendizaje automático Bosques Aleatorios (RF), Árboles de Decisión Alternativa (ADTree) y Función Discriminante Linear de Fisher (FLDA), las cuales fueron evaluadas mediante el área bajo la curva ROC dando como resultado de 0.89 para ADTree, 0.92 para FLDA y 0.97 para FR.

Zhou et al. (2018) realizaron un modelamiento de la susceptibilidad a deslizamientos mediante métodos de aprendizaje automático y métodos estadísticos multivariado en el área bordeada por la presa Las Tres Gargantas ubicadas entre las provincias de Chongqing y Hubei. China. Entre los métodos de automático utilizado están Máquina de Soporte Vectorial (SVM) y Redes de Neuronas Artificiales (ANN) y el método estadístico multivariado se encuentra la Regresión Logística (LR). El rendimiento de los modelos fue verificado mediante el área bajo la curva ROC la cual indicó que el modelo SVM tiene un AUC de 0.937, AAN con AUC de 0.868 y LR con AUC de 0.757 con la cual concluyeron que el modelo SVM aplicado fue el ideal para el área de estudio.

Como se ha podido apreciar en los estudios mencionados, en el Perú no se ha realizado el estudio de la susceptibilidad a deslizamientos mediante el uso de modelos de aprendizaje supervisado como sí se vienen realizando en otros países y demostrando que la aplicación de esta ciencia tiene buenos resultados.

2.2. Objetivos

2.2.1. Objetivo General

Estimar espacialmente la susceptibilidad de movimientos en masa mediante aprendizaje supervisado en la sub cuenca checras.

2.2.2. Objetivos Específicos

a. Conocer las potencialidades de los modelos de aprendizaje supervisado en el análisis espacial para la solución de los problemas del territorio y el ambiente.
b. Determinar la eficiencia de los métodos máquina de soporte vectorial y bosques aleatorios de aprendizaje supervisado en la selección de los modelos óptimos en la susceptibilidad a movimientos en masa.
c. Calcular el grado de probabilidad a movimientos en masa en la sub cuenca Checras mediante la aplicación de métodos de aprendizaje supervisado.

2.3. Zona de estudio

La presente investigación se enfocó en la sub cuenca Checras la cual está situado en la cabecera de la cuenca Huaura, cuenta con un área de 817.47 Km2 y se encuentra ubicada en los distritos de Checras, Pachangara, Oyon y Santa Leonor, provincia Huaura y Oyon y departamento de Lima.

2.4. Materiales

2.4.1. Software

Para los procesamientos de los datos e insumos se usaron softwares de libre acceso.

2.4.1.1. Python. Es un lenguaje de programación de acceso libre, de propósito general y orientado a objetos. Es un lenguaje muy usado en el desarrollo de modelos de aprendizaje automático, aprendizaje profundo e inteligencia artificial. En la presente investigación, para la implementación de estos modelos, se usaron diversos paquetes como:

A. Numpy. Numpy (*Numerical Python*) es un paquete muy utilizado en la computación científica. Su uso es principalmente en la manipulación de matrices multidimensionales.

B. Pandas: Es un paquete muy potente para la manipulación de datos y para el análisis de datos mediante el uso de *dataframe* en Python.

C. GDAL: El paquete GDAL es usado para la manipulación de datos geoespaciales como los de tipo ráster y vectoriales.

D. Matplotlib: Este paquete es muy utilizado para la creación y visualización de gráficos estadísticos, gráficos interactivos y animados.

E. SciKit-Learn: Paquete con una gran cantidad de funciones usado para la creación de modelos de aprendizaje automático tanto para modelos de aprendizaje supervisado como para modelos de aprendizaje no supervisado.

2.4.1.2. QGIS. Es un software de código abierto impulsado por voluntarios y es un proyecto de *Open Source Geospatial Foundation* (OSGeo). Es un software de Sistemas de Información Geográfica (SIG) desarrollado para el procesamiento de datos vectoriales, ráster y base de datos.

2.5. Metodología

2.5.1. Variables

Las variables utilizadas en la presente investigación se muestran en la siguiente tabla.

Tabla 1

Variables

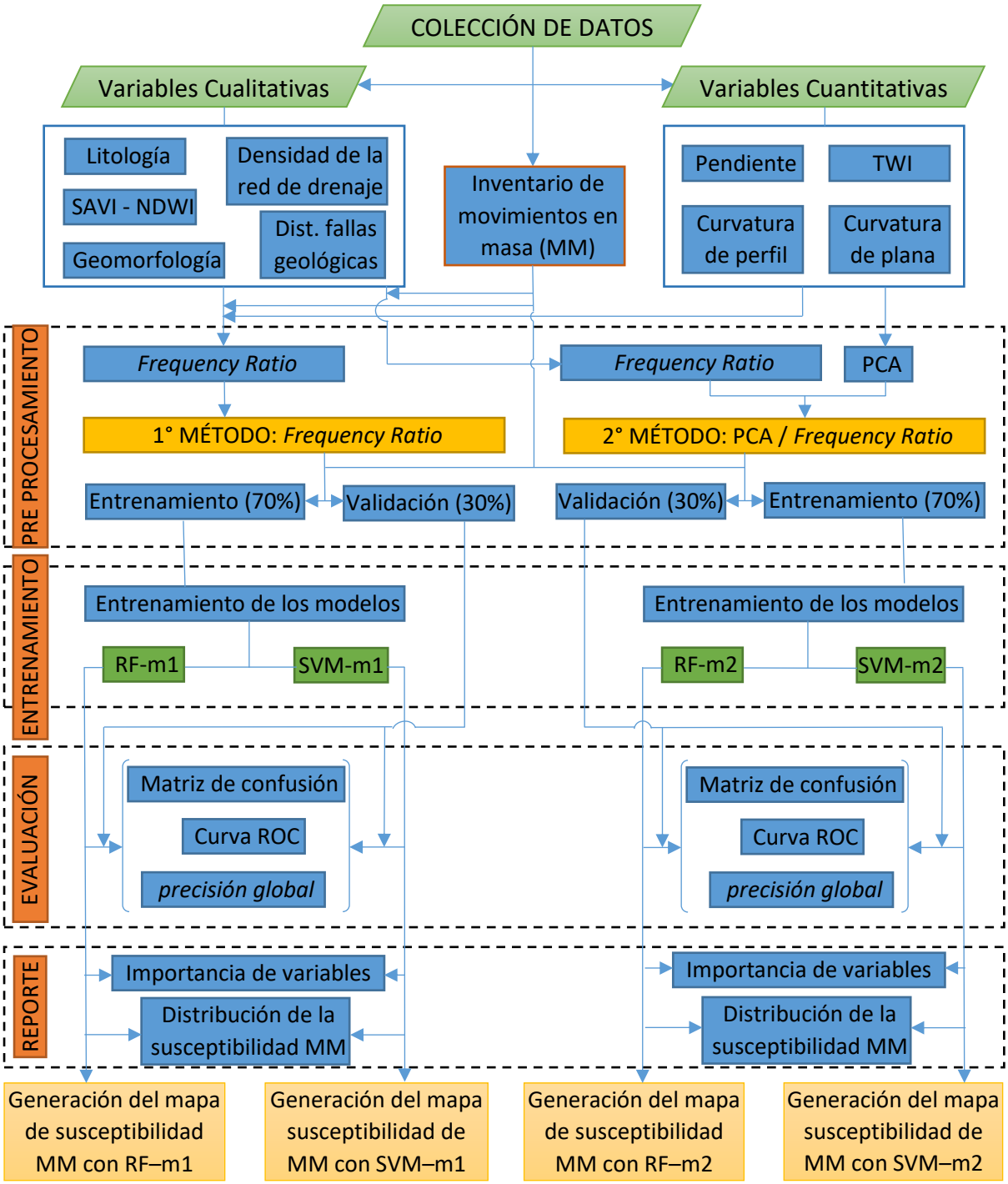
Variable Dependiente	Variable Independiente		
Movimientos en masa	Factores condicionantes	Cualitativas	Litología
			Distancia a las fallas geológicas
			Geomorfología
			Índice de vegetación ajustado al suelo – Índice diferencial normalizado de agua (SAVI - NDWI)
			Densidad de la red de drenaje
		Cuantitativas	Pendiente
			Índice de Topográfica Humedad (TWI)
			Curvatura de Perfil
			Curvatura plana

2.5.2. Metodología para procesamiento de los modelos

Para el pre procesamiento de los datos, entrenamiento y evaluación de los modelos se usó el lenguaje de programación Python y se desarrolló una metodología como se explica en el siguiente diagrama.

Diagrama 1

Flujo de trabajo de los modelos de aprendizaje supervisado



2.5.3. Principios básicos de aprendizaje automático supervisado

2.5.3.1. Definición. La traducción exacta de *Machine Learning* (ML) es aprendizaje de máquina o aprendizaje automático y esto nos hace inferir que el ML se refiere a que las máquinas adquieren conocimiento para un propósito. Raschka (2019) refiere que el aprendizaje automático ayuda a encontrar sentido y detectar patrones en los datos para transformar estos datos en conocimiento para, coincidiendo con el autor anterior, la predicción y toma de decisiones de futuros eventos.

2.5.3.2. Tipos. Dangeti (2017) divide el ML en tres diferentes tipos de aprendizaje fundamentales:

A. Aprendizaje supervisado. Este tipo de aprendizaje enseña a los algoritmos la relación que tienen las variables predictoras con la variable objetivo la cual tiene todas las clases a obtener. Según el tipo de clase abordado el aprendizaje supervisado se divide en dos grandes grupos.

Clasificación

La clasificación nos permite clasificar realizando predicciones sobre clases de atributo categórico. Como ejemplos se puede mencionar algunos modelos como bosques aleatorios, máquina de soporte vectorial y redes neuronales.

Regresión

La regresión es utilizada sobre clases de tipo continuo para la predicción de datos que no están enmarcados dentro de una cantidad predeterminada, a partir de las variables predictoras.

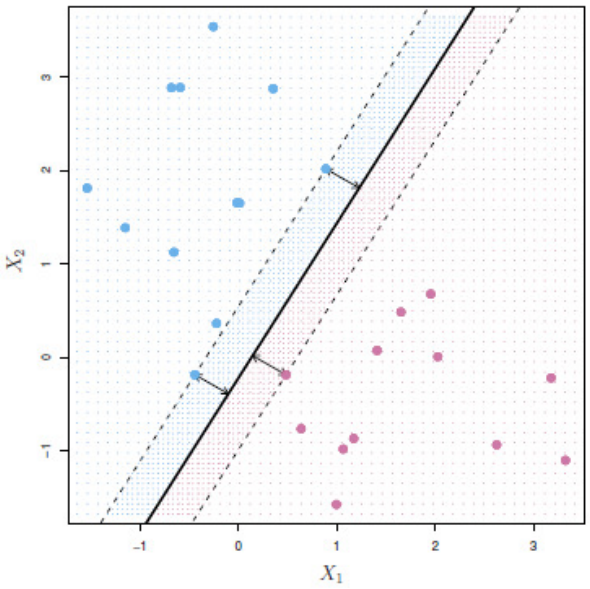
2.5.3.3. Modelos

A. Máquina de Soporte Vectorial (SVM). Máquina de Soporte Vectorial o *Support Vector Machine* es un modelo de aprendizaje automático desarrollado por Vladimir Vapnik y sus colaboradores en los noventas. Este modelo está diseñado para problemas tanto de regresión como de clasificación.

Dangeti (2017) explica que los SVM son modelos que funcionan mediante la maximización de los límites entre las características de los datos representados en un espacio multidimensional como se muestra en la figura 2. Este límite es llamado hiperplano y separa los datos lo más homogéneas posible.

Figura 2

Forma general del modelo Máquina de Soporte Vectorial



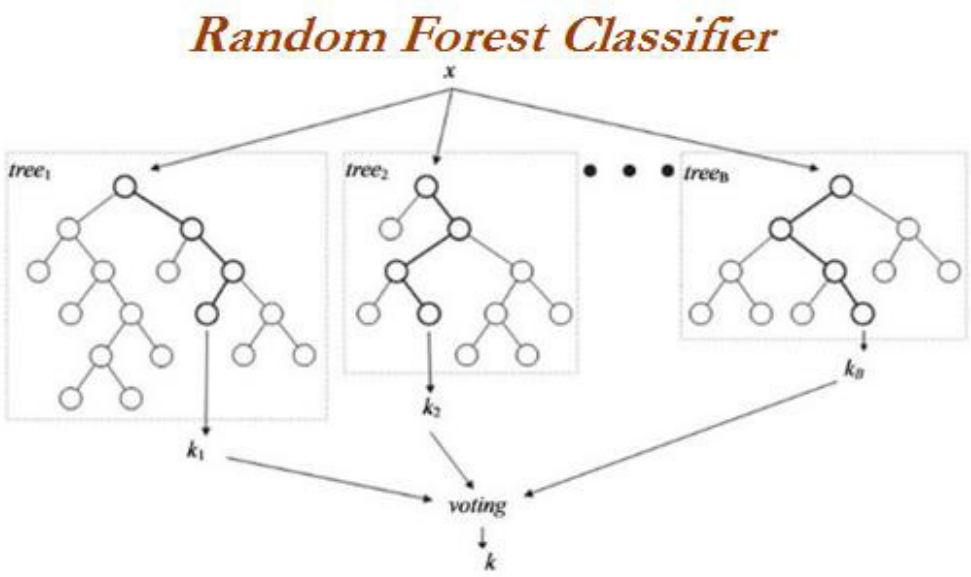
Nota. Tomado de *An Introduction to Statistical Learning* (p. 372), por G. James, 2021, Springer.

B. Bosque Aleatorio (RF). Bosques Aleatorios o *Random forest* es un modelo supervisado de aprendizaje automático que conforma tanto de clasificación como de regresión creado por Leo Breiman (2001) y su colaboradora Adele Cutler (Pita, 2017).

Este modelo está formado por el ensamble de muchos árboles de decisión como se muestra en la Figura 3, las cuales están formados por porciones aleatorias de las observaciones y características como se muestra en la figura 4, mediante el *bagging* o *bootstrap aggregation* la cual es una técnica de ensamblaje que reduce la correlación así reduciendo la varianza que se presentan en los árboles de decisión.

Figura 3

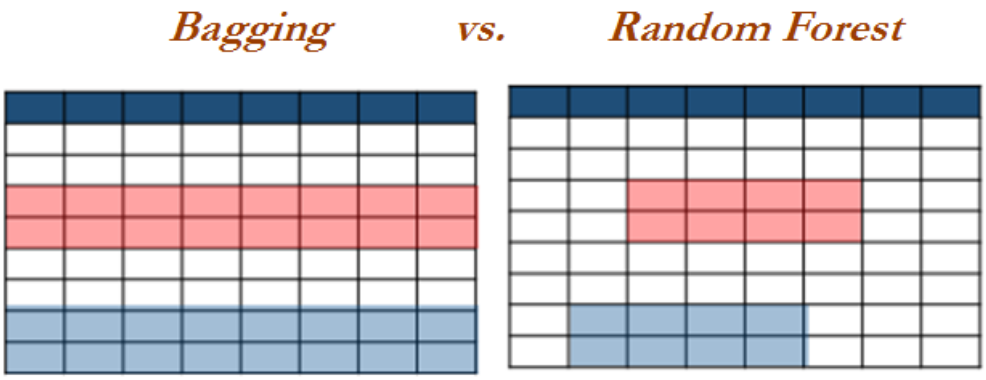
Ejemplo gráfico de clasificación por Bosques Aleatorios



Nota. Tomado de *Statistics for Machine Learning* (p. 112), por P. Dangeti, 2017, Packt.

Figura 4

Diferencia entre Baggin y Random Forest



Nota. Tomado de *Statistics for Machine Learning* (p. 112), por P. Dangeti, 2017, Packt.

C. *Frequency Ratio* (FR). *Frequency Ratio* es un modelo estadístico bivariado en la cual, para el caso del análisis de susceptibilidad a movimientos en masa, se determina la proporción de los movimientos en masa ocurridos y las clases de los factores considerados (Lee, 2014). Este modelo se define mediante la siguiente ecuación.

Ecuación 1

Frequency Ratio

$$FR = \frac{(A/B)}{(C/D)}$$

En la cual A es la cantidad de pixeles de movimientos en masa de una clase de un factor, B es el número total de pixeles de movimientos en masa en toda el área de estudio, C es el número total de pixeles de cada clase de cada factor y D es la cantidad total de pixeles del área de estudio.

Valores mayores a 1 indica que existe una alta correlación de las clases con la ocurrencia movimientos en masa, así, a medida que el valor FR disminuye, también, la correlación disminuye hasta el 0 que indica una inexistente relación entre ellas.

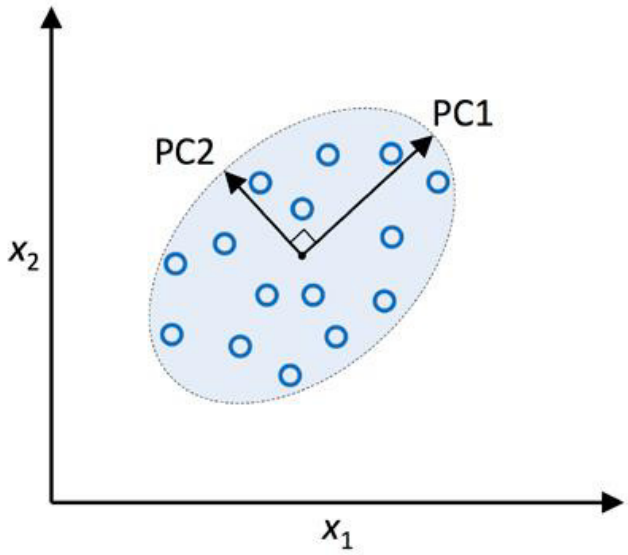
D. Análisis de componentes principales (PCA). El análisis de componentes principales (PCA por su sigla en inglés) es un modelo de aprendizaje automático no supervisado que es usado para la extracción de características, reducción de la dimensionalidad y tiene muchas utilidades como.

- Comprimir datos.
- Uso en el pre procesamiento de modelos supervisados.
- Reducción de la dimensión de los datos para su mejor comprensión, por ejemplo, en datos de muchas dimensiones reducir a dos dimensiones o tres dimensiones ayuda a una mejor visualización.

La interpretación de la máxima varianza de PCA permite hallar la dirección de los componentes principales, las cuales tienen la varianza máxima y son ortogonales (no correlacionados) entre sí, y los datos son proyectados a este nuevo sub espacio que tiene igual o menor dimensión que los datos originales como se muestra en la siguiente figura.

Figura 5

Análisis de componentes principales



Nota. Tomado de *Python Machine Learning* (p. 146), por S. Raschka, 2019, Packt.

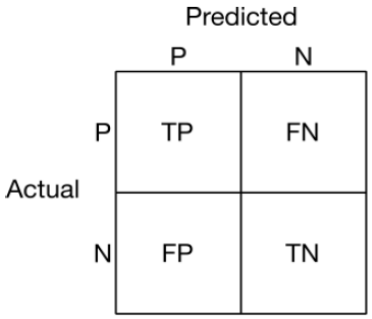
En la figura se muestra X_1 y X_2 las cuales son las características originales de los datos y PC1 y PC2 son los componentes principales en las cuales se captura la máxima varianza. Debido a que el primer componente principal tiene la mayor variabilidad y contenga la mayor información, se pueden representar los datos originales solo con este componente, así reduciendo la dimensionalidad de dos dimensiones a solo una dimensión.

2.5.3.4. Métricas de evaluación

A. Matriz de confusión. La matriz de confusión es una métrica de evaluación que nos muestra el desempeño de un modelo mediante un versus de los datos predichos y los datos originales o reales.

Figura 6

Matriz de confusión



Nota. Tomado de *Machine Learning with Python for Everyone* (p. 166)*,* por M. Fenner, 2019, Pearson.

- **Verdadero positivo (TP)** En el caso del presente trabajo de investigación, los verdaderos positivos son los casos en los que el modelo predice como movimientos en masa y, también, son movimientos en masa en la realidad.
- **Verdadero negativo (TN)** Los verdaderos negativos son los casos en el que el modelo predice como no movimiento en masa y en la realidad, también, no es movimiento en masa.

- **Falso positivo (FP)** Los falsos positivos son los casos en el que el modelo predice como movimiento en masa, pero en realidad no es movimiento en masa.
- **Falso negativo (FN)** Los falsos negativos son los casos en el que el modelo predice como no movimiento en masa, pero en realidad si hay un caso de movimiento en masa.

B. Precisión global. Es una métrica de evaluación de un modelo de clasificación para evaluar su desempeño. Nos indica qué tanto de las observaciones han sido predichas correctamente con respecto al total y se calcula mediante la siguiente ecuación.

Ecuación 2

Precisión global

$$ACC = \frac{TP + TN}{FP + FN + TP + TN}$$

Además, esta métrica es usada como un indicador para la determinación del sobre ajuste o sub ajuste de un modelo. Como indica Dangeti (2017), en qué momento se debe detener la afinación del modelo con respecto a sus hiperparámetros, se debe considerar estos tres estados.

Primer estado: Estado de sub ajuste, se presenta cuando se obtiene un bajo precisión global en el entrenamiento y un bajo precisión global en la prueba.

Segundo estado: Estado óptimo, se presenta cuando se obtiene un alto precisión global en el entrenamiento y, también, un alto precisión global en la prueba.

Tercer estado: Estado de sobre ajuste, se presenta cuando se obtiene un alto precisión global en el entrenamiento y un bajo precisión global en la prueba.

C. Área bajo la curva ROC. Característica operativa del receptor (ROC por sus siglas en inglés) es una métrica útil para la selección del modelo de clasificación según su desempeño. Esta métrica está basada en la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) calculados mediante el cambio del umbral de las clasificaciones como se muestra en la Figura 7.

La TPR y FPR se calculan tomando como base la matriz de confusión y se describen mediante las siguientes ecuaciones.

Ecuación 3

Tasa de verdaderos positivos (TPR)

$$TPR = \frac{TP}{TP + FN}$$

Ecuación 4

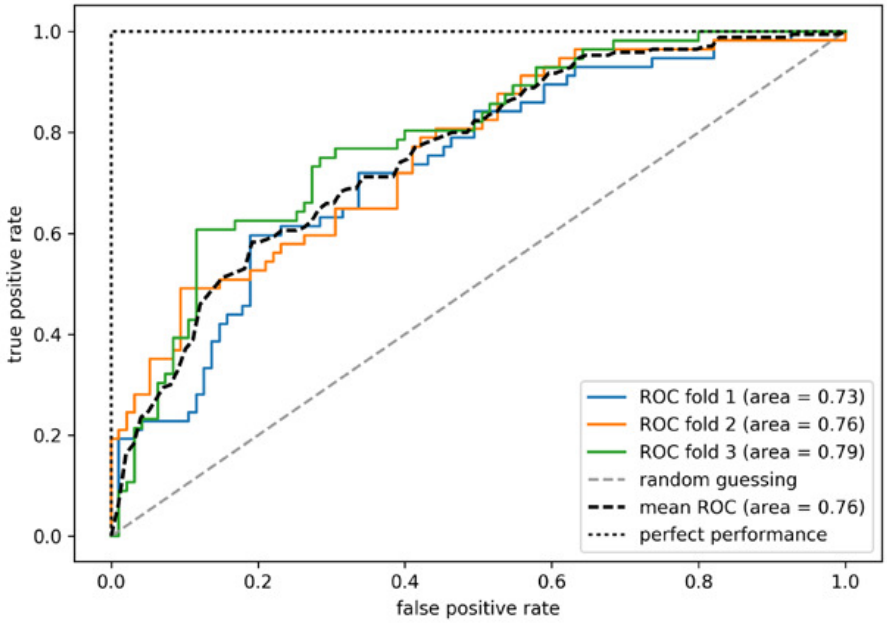
Tasa de falsos positivos (FPR)

$$FPR = \frac{FP}{FP + TN}$$

Raschka (2019) explica que los modelos que se encuentren más cercanos a la parte superior izquierda o área bajo la curva ROC más cercanos a uno, serán los modelos con mejor rendimiento que los que se encuentran más cercanos a la línea diagonal o por debajo de ella.

Figura 7

Curva ROC



Nota. Tomado de *Python Machine Learning* (p. 218)*,* por S. Raschka, 2019, Packt.

2.5.3.5. Importancia de las variables

A. Importancia de las variables Máquina de Soporte Vectorial. En la ecuación 1 se muestra la ecuación del hiperplano para un espacio bidimensional en las cuales β_1 y β_2 son los coeficientes o pesos. Bakharia, A. (1 de febrero del 2016), explica que la importancia de las variables en vectores de soporte vectorial se considera a partir de estos pesos que se obtienen de la ecuación del hiperplano formado. Además, la dirección indica la clase predicha.

B. Importancia de las variables Bosques Aleatorios. Para describir las variables que influyeron en mayor o menor medida que otros se utiliza la importancia de las variables, la cual es una lista en la que se ordenan las variables de acuerdo a cuánto se reduce la impureza media de los nodos de los árboles (Géron, 2019). En este caso se estará usando la medida de selección de atributos Gini.

2.6. Procedimiento

2.6.1. Procesamiento de los datos de movimientos en masa y factores condicionantes

Para el desarrollo de los objetivos es necesario contar con los registros de movimientos en masa, no movimientos en masa y factores condicionantes de la zona de estudio detallados en la tabla 2. Por este motivo los movimientos en masa se obtuvieron del procesamiento de la información de movimientos en masa obtenida del GEOCATMIN y se complementó dicha información mediante la fotointerpretación de imágenes de alta resolución y multitemporales de Google Earth e imágenes del satélite Sentinel 2. Para los factores condicionantes se procesaron los datos de carta geológica nacional para la obtención de la geología (anexo A) y fallas geológicas (anexo B); mapa geomorfológico del Perú para la obtención de la geomorfología (anexo C), imágenes satelitales para la cobertura vegetal; modelo digital del terreno para la densidad de drenaje (anexo D), pendiente (anexo E), índice Topográfica de Humedad (anexo F), curvatura de perfil (anexo G) y curvatura plana (anexo H).

2.6.2. Colección de los datos

Para el uso de los datos en el lenguaje de programación Python se siguió los siguientes pasos:

- Se usó las librerías necesarias para la manipulación de los factores condicionantes y los datos de movimientos en masa, así como *gdal*, librería que nos ayuda a la lectura de datos ráster, *pandas*, librería que permite la manipulación de *dataframes*.
- Los factores condicionantes y los datos de movimientos en masa se leyeron y se convirtieron en *dataframe* para su mejor manipulación, en filas donde se encuentran las observaciones y en columnas las variables, como se muestra en la sección A del código

general que se encuentra en el anexo I. Al *dataframe* se le asignaron los rangos de cada variable (figura 8).

Figura 8

Dataframe de las variables

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM
0	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Fuerte	Concava	Concava	Muy alto	Data a predecir
1	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir
1466	RCE-rs	Vegetación moderada	400-800 m	Media baja	Ki-ca	Baja	Concava	Concava	Alto	Sin MM
1775	RCE-rs	Suelo desnudo	>1000 m	Media baja	Ki-chu	Media	Concava	Concava	Alto	Sin MM
5272	RCE-rs	Suelo desnudo	>1000 m	Media baja	Ki-chu	Fuerte	Linear	Linear	Alto	MM
8325	RME-rs	Suelo desnudo	200-400 m	Baja	Ki-chi	Muy fuerte	Convexa	Convexa	Alto	MM

- Como se consideró las variables cualitativas y las variables cuantitativas por separado, es esencial conocer su estadística descriptiva para entender su distribución. Para las variables cualitativas se obtuvo el número de clases, la moda y la frecuencia de la moda y para las variables cuantitativas la media, la desviación estándar, percentiles, el valor mínimo y máximo.

Figura 9

Estadística descriptiva de las variables cualitativas

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia
count	912878	912878	912878	912878	912878
unique	17	5	5	5	15
top	RM-rs	Suelo desnudo	>1000 m	Media	PN-c
freq	357463	620991	325532	239655	228869

Figura 10

Estadística descriptiva de las variables cuantitativas

	Pendiente	Curvatura de perfil	Curvatura plana	TWI
count	912878.000000	912878.000000	912878.000000	912878.000000
mean	27.078819	-0.000144	0.000219	6.031451
std	11.238954	0.003824	0.071339	1.905293
min	0.000000	-0.047504	-20.824295	1.981781
25%	19.311764	-0.001984	-0.006332	4.791482
50%	27.827928	-0.000267	-0.000157	5.673806
75%	34.893112	0.001492	0.006106	6.798536
max	75.527184	0.078057	21.083567	22.946489

2.6.3. Pre procesamiento. En el pre procesamiento de los datos se consideró dos métodos en la preparación de los factores condicionantes. En el primer método se usó Frequency Ratio para todos los factores condicionantes, tanto para las cualitativas y para las cuantitativas, y en el segundo método se usó Frequency Ratio para las cualitativas y Análisis de Componentes Principales para las cuantitativas, como se muestra en la sección B y C del código general que se encuentra en el anexo I.

2.6.4. Entrenamiento. En el entrenamiento de los modelos, como se muestra en la sección D y E del código general que se encuentra en el anexo I, se siguió los siguientes pasos:

- Los datos se dividieron en dos porciones, con 70% para el entrenamiento del modelo y 30% para la evaluación del modelo, tanto para el primer método como para el segundo método.
- Se entrenaron los modelos de Máquina de Soporte Vectorial y Bosques Aleatorios para ambos métodos con la porción de los datos de entrenamiento y considerando los hiperparámetros más óptimos para cada modelo.

2.6.4.1. Hiperparámetros

A. Hiperparámetros del modelo Máquina de Soporte Vectorial. En esta sección se mencionarán los hiperparámetros más importantes usado para la realización del trabajo de investigación.

- **C.** Este hiperparámetro fue explicado en la sección de clasificador de soporte vectorial. En valores grandes de C, hay un mayor ajuste en el entrenamiento con mayor sobreajuste y fronteras de decisión más complejas lo que conlleva a un mayor sesgo, en contraste, valores pequeños de C, nos proporciona una menor exactitud, menor sobreajuste y fronteras de decisión menos complejas lo que conlleva a una mayor varianza.
- **Kernel.** El kernel para la clasificación de vectores de soporte es el "linear" y para la clasificación de observaciones no lineales son el "rbf" y el "poly", también, explicados en la sección anterior.

B. Hiperparámetros de Bosques Aleatorios. Para la construcción de los modelos se presentarán los hiperparámetros más relevantes del paquete proporcionado por SciKit – Learn.

Géron (2019), menciona que los hiperparámetros que se usan en bosques aleatorios son los mismos que se usan en la clasificación por árboles de decisión y clasificación por *baggin.*

- **n_stimator.** Es el número de árboles que se debe considerar en el bosque. Géron (2019) recomienda que sea lo mayor posible dado que a mayor cantidad de árboles aumentaría la robustez del ensamble y así reducir el sobreajuste. Sin embargo, se recomienda limitar la cantidad de árboles a la capacidad del equipo de cómputo y al tiempo que conlleva aplicar el modelo.

- **max_depth.** Es la profundidad máxima de un árbol de decisión. Se asigna un número entero para que el árbol se expanda hasta una cierta decisión, de lo contrario el árbol se profundiza hasta que sus hojas sean completamente puros. Un beneficio de restringir la profundidad del árbol es reducir el sobreajuste (Géron, 2019).
- **max_features.** Es el número de características que se va tomar para cada árbol de decisión. Considerar un número mínimo ayuda a disminuir el sobreajuste (Géron, 2019). En el caso de clasificación en bosques aleatorios se utiliza la raíz cuadrada del número total de las características.
- **min_samples_leaf.** Es la cantidad mínima de observaciones que requiere un nodo hijo para que haya una división.
- **Criterion.** Son las medidas de selección de atributos para medir la calidad de la división de nodos en los árboles de decisión. Para el caso de SciKit-Learn considera los tipos a) Gini y b) entropía.

2.6.5. Evaluación

La evaluación de los modelos consistió en averiguar el desempeño de estos, tanto para los modelos del primer método como para los modelos del segundo método, como se muestra en la sección D y E del código general que se encuentra en el anexo I.

2.6.6. Reporte

Para el reporte de los resultados del modelo se consideró la contribución de las variables en el entrenamiento de cada modelo y la distribución de la susceptibilidad a movimientos en masa en la totalidad de la sub cuenca, como se muestra en la sección D y E del código general que se encuentran en el anexo I.

2.6.6.1.Importancia de las variables. La importancia de las variables se determinó para cada modelo generado en el que se describe la influencia de cada variable en la

determinación de la susceptibilidad a movimientos en masa, como se muestra en la sección D y E del anexo I.

2.6.6.2. Distribución de la susceptibilidad a movimientos en masa. La distribución de los movimientos en masa en la totalidad de la cuenca se generó con la probabilidad de la predicción de la susceptibilidad mediante los modelos ya entrenados aplicados en los datos totales de la cuenca. La distribución se clasificó mediante el método de clasificación de cortes naturales (Jenks) considerando cinco clases, con la cual se distribuyó las probabilidades en susceptibilidad muy baja, baja, media, alta, muy alta como se muestra en la sección D y E del anexo I.

2.6.7. Generación del mapa de susceptibilidad a movimientos en masa. Para la generación de los mapas de susceptibilidad a movimientos en masa de cada uno de los modelos considerando los dos métodos, se utilizó la probabilidad de la predicción de la susceptibilidad, y considerando las propiedades del ráster de la cuenca se exportó un ráster en formato .tif, como se muestra en la sección D y E del código general que se encuentran en el anexo I. Para la presentación del mapa, que se realizó en Qgis, se utilizó la clasificación por cortes naturales (Jenks) que se consideró en la distribución de la susceptibilidad a movimientos en masa.

III. APORTES MÁS DESTACABLES A LA INSTITUCIÓN

3.1. Resultados

Una vez desarrollado los modelos que ayudarán a la obtención de la predicción de la susceptibilidad a movimientos en masa, se generó los mapas de la susceptibilidad utilizando los modelos descritos. El desarrollo de estos modelos, en general, nos generó los siguientes resultados.

3.1.1. Métodos de pre procesamiento de los factores

Como se explicó anteriormente, el pre procesamiento se dividió en dos métodos, primero, el método donde se obtuvo el valor de *Frequency Ratio* de todos los factores y el segundo en la que se hace una combinación de *Frequency Ratio* para los factores con variables cualitativas y el Análisis de componentes principales para factores con variables cuantitativas.

3.1.1.1. Frequency Ratio. Los valores de FR se obtuvieron mediante la ecuación 1 de la cual se obtienen valores de cero a mayores a uno la que indica el grado de correlación que tienen las clases de los factores con los movimientos en masa.

En la Tabla 3 se aprecia que en el factor litología, las clases pórfido cuarcífero (KP-pcz) y depósitos aluviales (Qh-al) son las clases que tienen un FR mayor a uno, indicando que son las zonas en las que existe una mayor susceptibilidad a movimientos en masa. Cobbing (1973), indica en el boletín geológico N° 26, que en los pórfidos cuarcíferos se encuentran rocas intrusivas que están altamente intemperizadas, y en los depósitos aluviales se encuentran acumulación de gravas asociadas con capas de arena, limo, arena arcillosa y con materiales intemperizados. Las distancias a las fallas se obtuvieron valores relativamente altos en todas sus clases, pero en las distancias de 200 – 400 m se obtuvo el valor más alto debido a que la proximidad a las fallas aumenta la ocurrencia de movimientos en masa. En la geomorfología se encontró dos clases que se diferencian de las demás por tener valores de FR muy altos como

son las sub unidades terraza indiferenciada (Ti) la cual está formada por terrazas de diferentes edades y la sub unidad Vertiente o piedemonte aluvio-torrencial (P-at) que están formados por flujos de detritos y lodos. En el factor SAVI-NDWI se encuentra solo la clase suelo desnudo que presenta un valor de FR mayor a uno evidenciando que la presencia de vegetación juega un rol muy importante en la ocurrencia de estos fenómenos de origen natural. El factor densidad de drenaje mantiene una correlación directa con el valor de FR en la cual los valores altos de este factor como son las clases media alta y alta presentan valores mayores a uno de FR. Al igual que la densidad de drenaje el factor pendiente presenta una relación directa con los valores de FR y presentando a las clases muy fuerte y abrupta, valores mayores a uno. El factor TWI, también, presenta una relación directa con FR y con las clases medio, alto y muy alto, con valores mayores que el resto, esto corrobora la teoría en la cual se indica que estos valores altos de TWI son propensas a acumulaciones de agua que contribuirían a la ocurrencia de los movimientos en masa. La curvatura de perfil y plana son dos factores que tienen influencia en el flujo, mientras que en la primera las clases linear y convexa presentan valores de FR más altos en la segunda las clases con mayor FR son la linear y la cóncava.

Tabla 2

Frequency Ratio de los factores considerados

Factor	Clases	#Pixeles MM	#Pixeles totales	%Pixeles MM	%Pixeles total	Frequency Ratio
Litología	**KP-dia**	0	75	0.00	0.01	0.00
	KP-pcz	3	1148	0.42	0.13	3.34
	Ki-ca	115	204651	16.11	22.42	0.72
	Ki-chi	223	167810	31.23	18.38	1.70
	Ki-chu	34	42934	4.76	4.70	1.01
	Ki-f	9	18215	1.26	2.00	0.63
	Ki-oy	4	3442	0.56	0.38	1.49
	Ki-ph	21	24114	2.94	2.64	1.11
	Ki-pt	34	52209	4.76	5.72	0.83
	Ki-s	52	54264	7.28	5.94	1.23
	Ks-ce	0	722	0.00	0.08	0.00

Tabla 2. *Continuación*

	PN-c	142	228869	19.89	25.07	0.79
	Q-gl	6	26416	0.84	2.89	0.29
	Qh-al	22	11096	3.08	1.22	2.53
Distancia fallas	**0-200 m**	107	140668	14.99	15.41	0.97
	200-400 m	107	132116	14.99	14.47	1.04
	400-800 m	175	222984	24.51	24.43	1.00
	800-1000 m	64	91578	8.96	10.03	0.89
	>1000 m	261	325532	36.55	35.66	1.03
Geomorfología	**Bo**	0	972	0.00	0.11	0.00
	Mo	6	13910	0.84	1.52	0.55
	P-at	16	7176	2.24	0.79	2.85
	RCE-rs	45	76576	6.30	8.39	0.75
	RCL-ri	0	1080	0.00	0.12	0.00
	RCL-rs	6	22554	0.84	2.47	0.34
	RCL-rv	5	5889	0.70	0.65	1.09
	RM-ri	15	16184	2.10	1.77	1.19
	RM-rs	273	357463	38.24	39.16	0.98
	RM-rv	65	95955	9.10	10.51	0.87
	RM-rvs	7	16377	0.98	1.79	0.55
	RMCE-rs	0	1610	0.00	0.18	0.00
	RME-rs	150	170732	21.01	18.70	1.12
	Ti	4	1699	0.56	0.19	3.01
	V-cd	77	53876	10.78	5.90	1.83
	V-d	17	16664	2.38	1.83	1.30
	Vll-gl	28	54161	3.92	5.93	0.66
SAVI_NDWI	**Agua**	0	6782	0.00	0.74	0.00
	Suelo desnudo	568	620991	79.55	68.03	1.17
	Vegetación dispersa	101	199878	14.15	21.90	0.65
	Vegetación moderada	41	72903	5.74	7.99	0.72
	Vegetación densa	4	12324	0.56	1.35	0.41
Densidad drenaje	**Baja**	123	197815	17.23	21.67	0.79
	Media baja	112	203256	15.69	22.27	0.70
	Media	178	239655	24.93	26.25	0.95
	Media alta	202	195152	28.29	21.38	1.32
	Alta	99	77000	13.87	8.43	1.64
Pendiente	**Muy baja**	0	3431	0.00	0.38	0.00
	Baja	4	21907	0.56	2.40	0.23
	Media	37	118864	5.18	13.02	0.40
	Fuerte	88	228313	12.32	25.01	0.49
	Muy fuerte	545	498127	76.33	54.57	1.40
	Abrupta	40	42236	5.60	4.63	1.21
TWI	**Muy bajo**	9	15590	1.26	1.71	0.74
	Bajo	30	69072	4.20	7.57	0.56
	Medio	149	207166	20.87	22.69	0.92

Tabla 2. *Continuación*

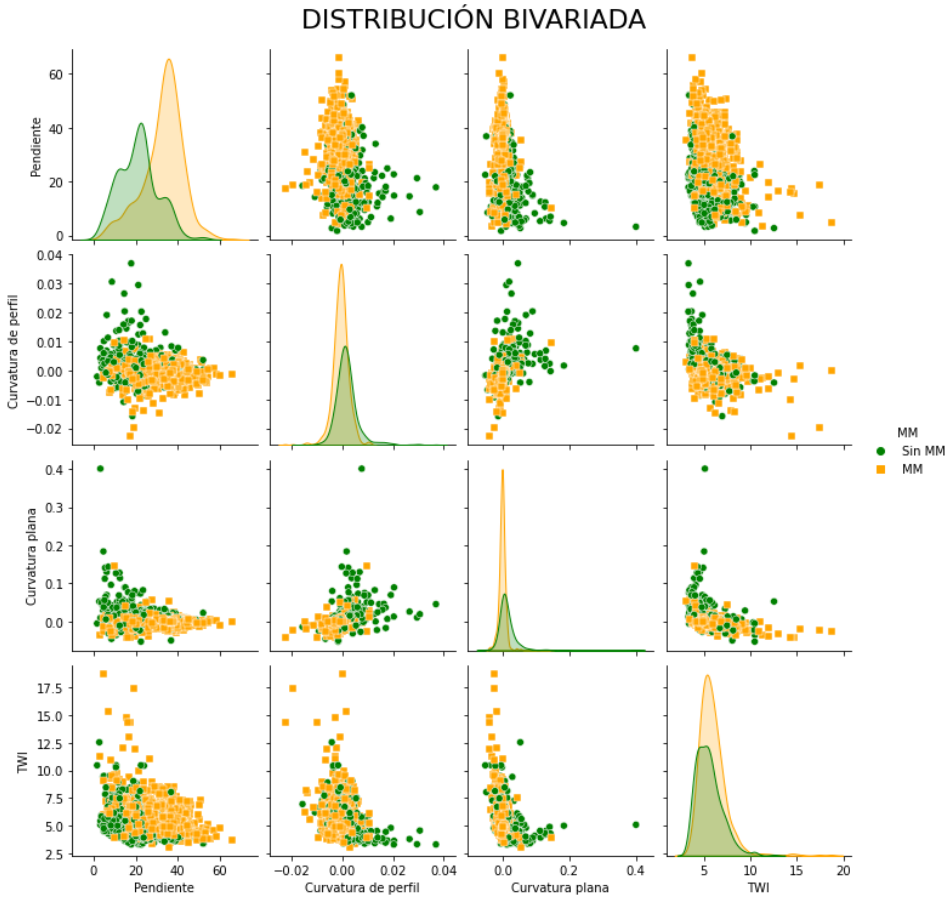
	Alto	333	337331	46.64	36.95	1.26
	Muy alto	193	283719	27.03	31.08	0.87
Curvatura de perfil	**Cóncava**	77	147423	10.78	16.15	0.67
	Linear	463	586217	64.85	64.22	1.01
	Convexa	174	179238	24.37	19.63	1.24
Curvatura plana	**Convexa**	69	184817	9.66	20.25	0.48
	Linear	490	540715	68.63	59.23	1.16
	Cóncava	155	187346	21.71	20.52	1.06

3.1.1.2. Análisis de Componentes Principales. Para el Análisis de Componentes Principales se realizó un análisis previo mediante la distribución bivariado y la matriz de correlación, las cuales se aplicaron en las variables cuantitativas como son la pendiente, TWI, curvatura de perfil y la curvatura plana. Este análisis previo se realizó con la finalidad de conocer el comportamiento uno a uno de los factores mencionados, además del comportamiento que tienen cada uno de ellos con los datos de movimientos en masa y los datos que no tienen movimientos en masa.

En el gráfico se observa una relación uno a uno (distribución bivariada) de las variables numéricas y en diagonal se observa el histograma de cada variable (distribución univariada). En el caso de la distribución bivariada los factores curvatura plana y curvatura de perfil se observa que tienen mayor correlación directa entre ellas y cada una de ellas una correlación inversa con el factor TWI, además, se observa que la pendiente mantiene una correlación inversa con la curvatura plana y la curvatura de perfil lo que nos indica que entre ellas podría haber información redundante. En el caso de la distribución univariada se aprecia en la pendiente que la información de movimientos en masa y sin movimientos en masa tiene un mayor grado de separabilidad por la diferencia en los picos que tienen en su histograma, lo que nos sugiere que este factor tiene una gran influencia en los modelos implementados.

Figura 11

Distribución bivariada de las variables cuantitativas

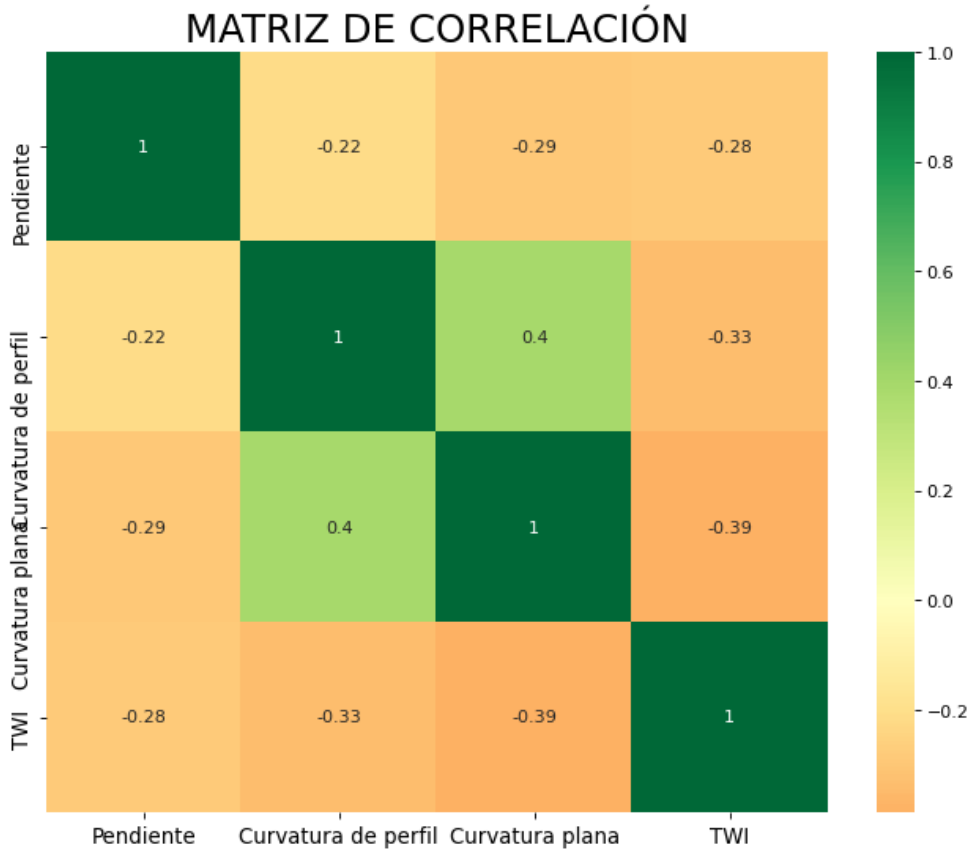


Para corroborar la correlación que existe entre las variables consideradas se tiene en cuenta un mapa de calor de la matriz de correlación en la cual se aprecia los valores del coeficiente de correlación de Pearson, la cual se encuentra en un rango de -1 a 1 indicando los valores extremos una gran correlación lineal y el valor de cero una correlación lineal nula.

En la figura se observa que las variables con mayor correlación lineal directa son la curvatura plana y la curvatura de perfil y con mayor correlación lineal inversa las variables curvatura plana y TWI, además, se aprecia que la curvatura de perfil y el TWI tienen una correlación lineal inversa considerable.

Figura 12

Matriz de correlación



Todo lo mencionado sugiere que esos factores tienen información redundante entre ellas. Para ello se usó el modelo de Análisis de Componentes Principales que ayudó a la reducción de la dimensionalidad y al mismo tiempo obtener información con menor redundancia.

Para iniciar el proceso se normalizó los datos según el escalado de máximo y mínimos que nos proporciona *sklearn* en la cual los datos de cada factor se normalizan en un rango de cero a uno.

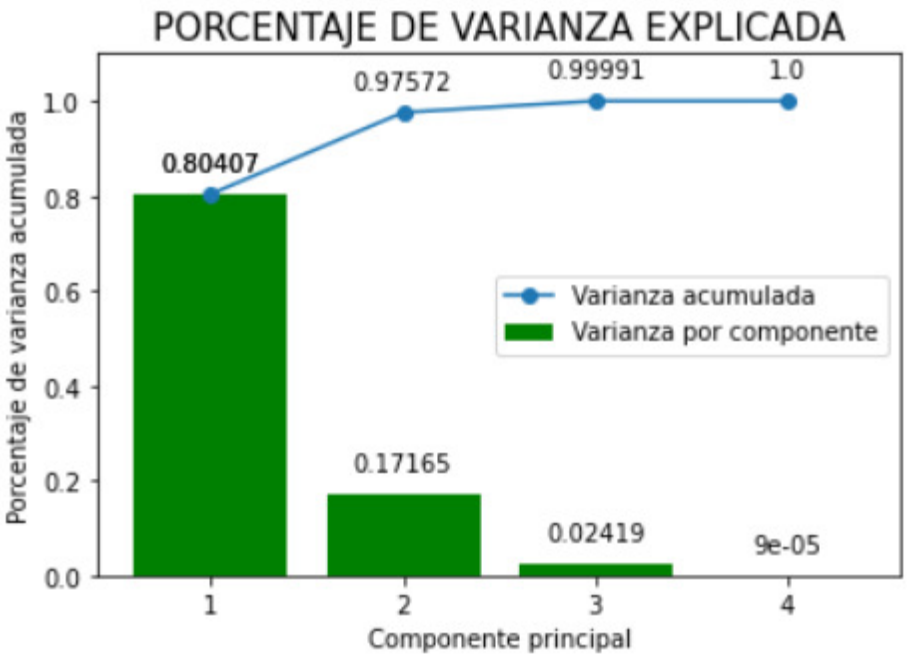
En los resultados del PCA se obtiene una cantidad de componentes principales (PC) tanto como el número de variables usados. Las 4 variables consideradas generaron cuatro componentes principales las cuales fueron analizadas mediante su varianza explicada de cada

una de ellas. En la siguiente Figura 13 se presenta las varianzas explicadas de cada componente y las varianzas explicadas acumuladas.

En la figura se observa el orden en que se encuentran cada componente según el valor de su varianza explicada. El primer componente explica un 80.41 % del total de los datos, el segundo explica un 17.17 %, el tercero un 2.41 % y el cuarto componente tan solo un 0.009 %, lo que significa que con tan solo los dos primeros componentes se explica el 97.58 % de los datos iniciales y en los tres primeros componentes está explicado el 99.99 % de los datos iniciales. Por esta razón se consideró, para la aplicación de los modelos, solo los tres primeros componentes principales, así descartando el tercer componente principal debido a que aporta una mínima porción del total de los datos.

Figura 13

Porcentaje de varianza explicada



3.1.1.3. Rendimiento de los modelos de predicción. Para la evaluación del rendimiento de un modelo se tomó en cuenta las métricas como la matriz de confusión, precisión global, área bajo la curva ROC las cuales se obtuvieron con los datos de validación, inmediatamente después del entrenamiento de cada modelo.

A. Modelos de predicción e hiperparámetros. Los hiperparámetros usados en el entrenamiento de los modelos determinan el rendimiento de cada una de estas, para lo cual se realizó el entrenamiento con distintos valores de hiperparámetros por prueba y error.

En el modelo de máquina de soporte vectorial los principales hiperparámetros considerados en ambos métodos son el parámetro *C* con un valor de 1.0 en ambos métodos debido a que las observaciones que se encuentran tienen valores muy cercanos a cero, y el kernel se consideró linear.

Tabla 3

Hiperparámetros del modelo SVM

Hiperparámetros	SVM	
	Primer método	Segundo método
C	1.0	1.0
Kernel	Linear	Linear

En el caso del modelo bosques aleatorios los principales hiperparámetros que se consideraron en ambos métodos fueron *n_stimator* con valor 50 lo que indica que se obtendrá un bosque con 50 árboles, *max_depth* con valor 5 para que cada uno de los árboles se expanda hasta esa profundidad, *min_samples_leaf* con valor de 8 para que los nodos no se dividan con una menor cantidad de observaciones a este valor, y *Criterion* con la medida *Gini*.

Tabla 4

Hiperparámetros del modelo RF

Hiperparámetros	RF	
	Primer método	Segundo método
n_stimator	50	50
max_depth	5	5
min_samples_leaf	8	8
Criterion	Gini	Gini

Los hiperparámetros restantes se consideraron por defecto del algoritmo que proporciona *sklearn.*

B. Matriz de confusión. La matriz de confusión es la primera métrica de evaluación del rendimiento de un modelo de clasificación que se consideró debido a que nos indica los verdaderos positivos, verdaderos negativos, falsos negativos y falsos positivos de la predicción que realiza el modelo.

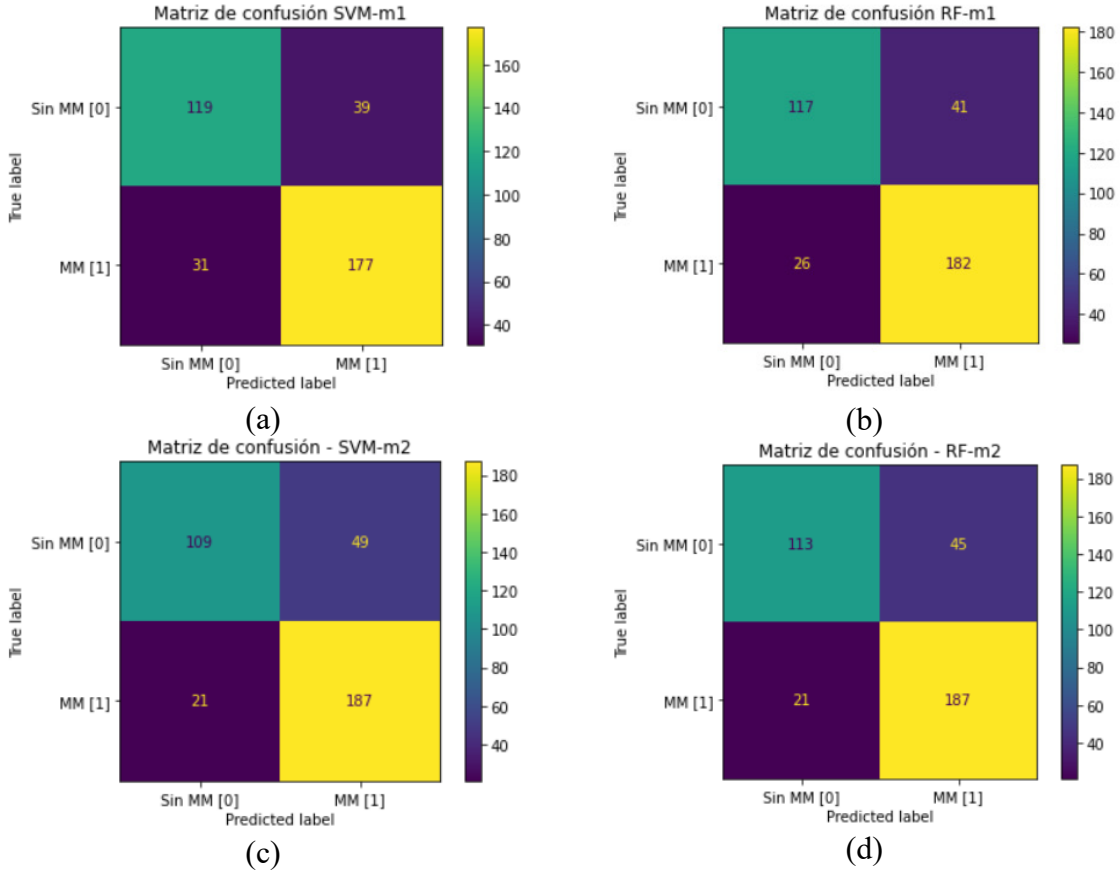
En escenarios ideales se desea que los falsos negativos y los falsos positivos sea cero, ya que así no tendríamos predicciones erróneas a la realidad, pero esto no es el caso de modelo entrenados con datos reales.

En la Figura 14 las matrices nos indican que los modelos aplicados a cada método nos proporcionan resultados buenos, ya que los falsos positivos y negativos son bajos y los verdaderos positivos y negativos son altos. En las matrices del primer método (a) y (b) se aprecia que el verdadero positivo del modelo RF es superior al del modelo SVM, lo que nos indica que el modelo RF predice mejor las observaciones donde sí existen movimientos en masa, mientras que en las matrices del segundo método (b) y (c) se observa que los verdaderos positivos tienen el mismo valor, pero el modelo RF supera en los valores de los verdaderos

negativos indicándonos que el modelo RF predice mucho mejor las observaciones donde no existen movimientos en masa. Entre los dos métodos considerados, los modelos que obtienen el mayor valor de verdaderos positivos son los del segundo método mientras que los modelos que obtienen los valores más altos en verdaderos negativos son los modelos del primer método, sugiriéndonos que los modelos de estos últimos ayudarían a predecir mejor las observaciones donde no podría ocurrir movimientos en masa, mientras que los modelos del segundo método nos ayudarían a predecir mejor las observaciones donde podría ocurrir movimientos en masa

Figura 14

Matriz de confusión de los modelos de predicción entrenados



.

C. Precisión global. La precisión global es una métrica de evaluación que nos proporcionaría una idea, como se explicó anteriormente, de si el modelo está sufriendo un sobre ajuste o un sub ajuste. Para esto es necesario obtener esta métrica de los datos de entrenamiento que fueron usados para el entrenamiento de los modelos y validación que son usados en la evaluación del rendimiento de los modelos como se muestra en la siguiente tabla 5.

En todas las medidas encontramos resultados superiores a los 0.79, lo que nos muestra que los modelos predicen correctamente mayor a 79% del total de los datos de entrenamiento. Los modelos RF de los dos métodos nos muestran valores superiores a los modelos SVM, indicándonos que estos modelos presentan un mejor desempeño, siendo el modelo RF del segundo método un tanto superior al del primer método.

Tabla 5

Precisión global de los modelos de predicción

Precisión global	Método 1		Método 2	
	SVM	RF	SVM	RF
Entrenamiento	0.7943	0.8366	0.7931	0.8472
Validación	0.8087	0.8169	0.8087	0.8196

Esta métrica nos proporciona una idea del sobre ajuste o sub ajuste que tiene el modelo mediante su evaluación en los datos de entrenamiento y validación como se indicó anteriormente. Todas las precisiones globales obtenidas tienen valores altos, lo que nos indica que los modelos de los dos métodos no presentan sub ajuste o sobreajuste y se encuentra en su estado óptimo.

D. Área bajo la curva ROC. El área bajo la curva ROC es una métrica de evaluación de modelos de clasificación que se utiliza esencialmente para comparar el rendimiento entre modelos y así determinar su eficiencia. En el caso del presente estudio se utilizó para

determinar el rendimiento de cada uno de los modelos que fueron usados en los dos métodos considerados como se muestra en la Figura 15 y 16.

En la figura 15, que representa el primer método, se observa que el modelo RF presenta un AUC de 0.900 y el modelo SVM 0.880 lo que nos indica que el modelo RF tiene un mejor rendimiento. En la figura 16, que representa el segundo método, se observa que el modelo RF posee un valor de AUC de 0.908 y mientras que el modelo SVM tiene 0.899 indicándonos que el modelo RF, al igual que en el primer método, presenta un mejor rendimiento. Comparando los modelos de los dos métodos, los que obtienen un mayor valor de AUC son los modelos del segundo método, pero la diferencia no es muy considerable.

Figura 15

Área bajo la curva ROC de los modelos del primer método

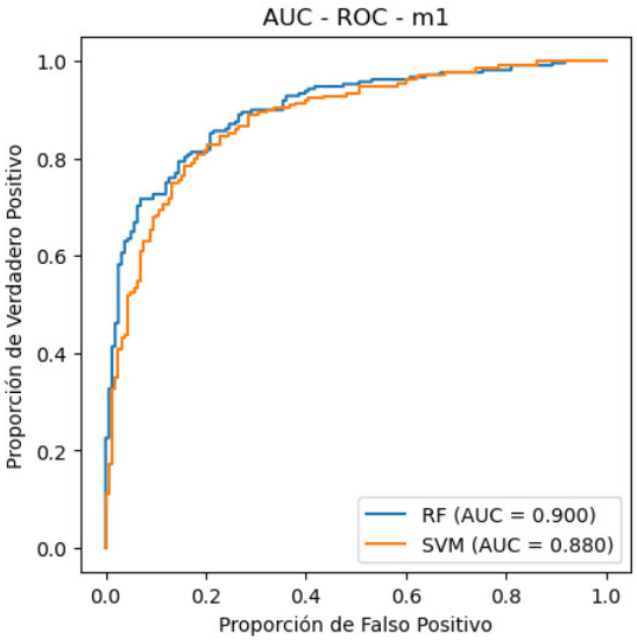
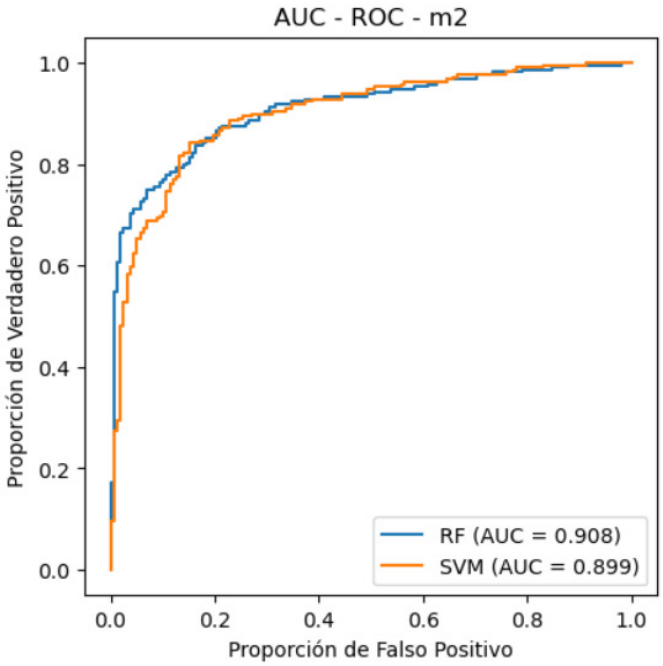


Figura 16

Área bajo la curva ROC de los modelos del segundo método



Todas las métricas de evaluación que se implementaron en los modelos de clasificación nos muestran que los modelos que se desarrollaron con el pre procesamiento del segundo método son los que presentan un mejor rendimiento en la predicción de los movimientos en masa y por consiguiente indicándonos que el pre procesamiento de este método es el más óptimo para la aplicación en estos tipos de datos.

3.1.1.4.Importancia de los factores en la susceptibilidad. La importancia de los factores está medida de acuerdo a la contribución que tienen los factores en el entrenamiento del modelo para predecir la presencia o ausencia de movimientos en masa. En el caso de los modelos utilizados, a mayor coeficiente mayor será su contribución en el entrenamiento del modelo, por el contrario, a menor valor del coeficiente su contribución será mínima.

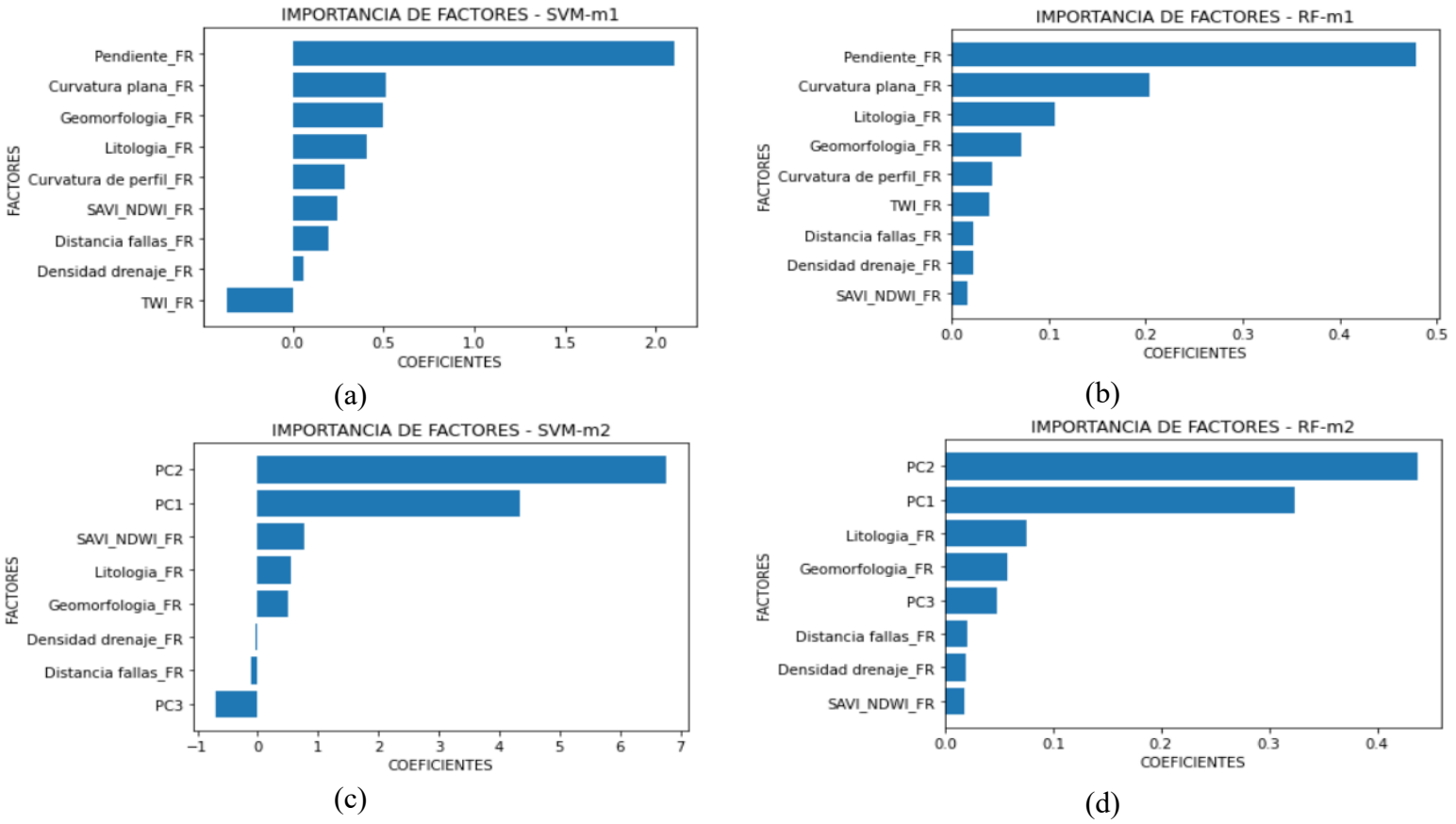
En las figuras se aprecia la importancia de los factores de los cuatro modelos implementados. En la importancia de los factores del modelo SVM del primer método (a) se

observa que los factores que presenta una gran influencia en predecir la presencia de movimientos en masa son la pendiente seguida por la curvatura plana, geomorfología, litología, curvatura de perfil, SAVI – NDVI, distancia a fallas y densidad de drenaje, por el contrario, para predecir la ausencia de movimientos en masa es el factor TWI. En el modelo RF del primer método (b), también, la pendiente tiene una gran influencia seguida de la curvatura plana, litología, geomorfología, curvatura de perfil, TWI, distancia a fallas, densidad de drenaje, SAVI – NDVI. En el modelo SVM del segundo método (c) se observa una gran influencia en predecir los movimientos en masa, el factor segundo componente principal seguido del primer componente principal, SAVI – NDWI, litología, geomorfología, y en predecir la ausencia de movimientos en masa el factor tercer componente principal, distancia a fallas y densidad de drenaje. El modelo RF del segundo método (d) el segundo componente principal tiene en conjunto con el primer componente principal tienen una gran influencia seguidos de la litología, geomorfología, el tercer componente principal, distancia a fallas, densidad de drenaje y SAVI – NDWI.

Se aprecia en los modelos del primer método que el factor que aporta una gran influencia en la predicción de los movimientos en masa es la pendiente seguida de la curvatura plana y en el segundo método es el segundo componente principal seguido del primer componente principal.

Figura 17

Importancia de los factores de los modelos predictivos



3.1.1.5. Distribución de la susceptibilidad a movimientos en masa. La distribución de la susceptibilidad a movimientos en masa, aplicados en la totalidad de los datos de la sub cuenca, se dividió en cinco clases Muy Baja, Baja, Media, Alta y Muy Alta, como se muestra en las Tablas 7 y 8 las cuales fueron clasificadas de acuerdo a la clasificación de cortes naturales (Jenks).

En la Tabla 6 y Figura 18 se observa que en el modelo SVM se clasificó en mayor porcentaje la clase Muy Alta con el 31.62 % del total del área de la sub cuenca, mientras que en menor porcentaje se clasificó la clase Media con 6.57 %. En el modelo RF, también, se clasifico en mayor porcentaje la clase Muy Alta con un porcentaje de 29.67 %, mientras que la clase con el menor porcentaje fue la clase Muy Baja con un porcentaje de 10.46 %. Se aprecia que las clases obtenidas tienen valores muy heterogéneos en el que se presentan valores muy bajos en algunas clases y valores altos en otras clases.

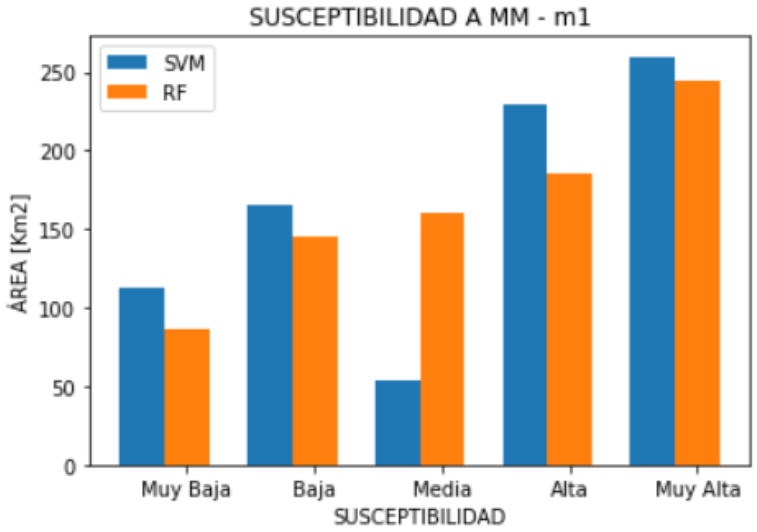
Tabla 6

Distribución de la susceptibilidad a movimientos en masa del primer método

Susceptibilidad	SVM		RF	
	Área [Km2]	Porcentaje	Área [Km2]	Porcentaje
Muy Baja	112.78	13.73	85.95	10.46
Baja	165.66	20.16	145.61	17.72
Media	53.96	6.57	160.91	19.58
Alta	229.42	27.92	185.34	22.56
Muy Alta	259.78	31.62	243.79	29.67

Figura 18

Distribución de la susceptibilidad a movimientos en masa del primer método



En el caso del segundo método las distribuciones de las clasificaciones de la susceptibilidad a movimientos en masa de los dos modelos son más homogéneos entre ellos como se muestra en la tabla 7 y figura 19. En el modelo SVM la clase con mayor porcentaje es la clase Alta con un porcentaje de 23.39 % y la clase con un menor porcentaje es la clase Muy Baja con 16.14 %. En el modelo RF la clase con mayor porcentaje es la clase Muy Alta con 22.37 % y la clase con menor porcentaje es la clase Muy Baja con 18.09 %.

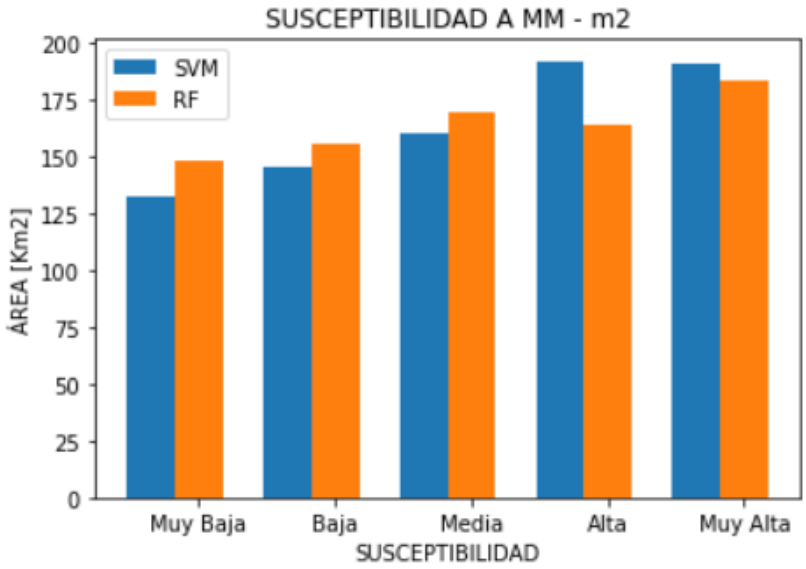
Tabla 7

Distribución de la susceptibilidad a movimientos en masa del segundo método

Susceptibilidad	SVM		RF	
	Área [Km2]	Porcentaje	Área [Km2]	Porcentaje
Muy Baja	132.63	16.14	148.62	18.09
Baja	145.56	17.72	155.91	18.98
Media	160.57	19.54	169.17	20.59
Alta	192.17	23.39	164.1	19.97
Muy Alta	190.67	23.21	183.79	22.37

Figura 19

Distribución de la susceptibilidad a movimientos en masa del segundo método



3.1.1.6. Estimación espacial de la susceptibilidad a movimientos en masa. La susceptibilidad a movimientos en masa del área de estudio se generó aplicando los modelos entrenados en los datos de toda la sub cuenca y se clasificó en cinco clases como se mencionó en la distribución de la susceptibilidad. Estos datos fueron plasmados en el mapa 1 para el modelo SVM del primer método, el mapa 2 para el modelo RF del primer método, en el mapa 3 para el modelo SVM del segundo método y en el mapa 4 para el modelo RF del segundo método mostrados en el anexo J.

A. Análisis e interpretación de la estimación espacial de la susceptibilidad de movimientos en masa. Para el análisis de los resultados obtenidos con los modelos aplicados a los dos métodos de pre procesamiento se escogieron dos áreas de análisis como se aprecia en las figuras 20 y 21. El área 1 se encuentra en la parte baja (oeste) de la sub cuenca, la cual se encuentra en las intersecciones del río Checras con las quebradas Pumapuchupan, Yuraccasha y Cayash, y el área 2 se encuentra en la parte (este) alta de la sub cuenca conformada por la parte superior de la quebrada Morococha.

Los resultados en las áreas 1 y 2 de los modelos del primer método, SVM en la figura 20 (a, c) y RF en la figura 48 (b, d), se aprecia que las zonas de muy alta susceptibilidad se encuentran en clases de muy fuerte y abrupta pendiente, además, coinciden mayormente con la clase cóncava del factor curvatura plana debido a que este factor tiene influencia en el flujo divergente y convergente. Las zonas de baja y muy baja susceptibilidad se encuentran coincidiendo con clases de baja pendiente y, además, coincidiendo en gran medida con la clasificación muy baja del factor TWI, como, también, se aprecia en la importancia de los factores del modelo SVM, a causa de que estos valores en este factor son zonas secas con buen drenaje. Los dos modelos de predicción presentan factores que ayudan a la disgregación de las clases de susceptibilidad, en el caso de RF se aprecia que la litología tiene una influencia moderada ya que ayuda al modelo a disgregar las clases, y en el caso de SVM se aprecia que el factor geomorfología ayuda a disgregar las clases de susceptibilidad.

Los resultados del segundo método en las áreas 1 y 2, SVM en la figura 49 (a, c) y RF en la figura 21 (b, d), y en los resultados de la importancia de variables nos muestran que los componentes principales uno y dos tiene una gran influencia en las zonas de susceptibilidad alta y muy alta, ya que se observa que valores altos de estos componentes principales se correlacionan con valores altos de susceptibilidad. Las zonas de baja susceptibilidad coinciden en gran medida con los valores bajos de los componentes principales dos y uno, además, según la importancia de las variables, estas zonas deberían guardar una relación con el tercer componente principal, la distancia a las fallas y en menor medida con la densidad de drenaje, pero no se observa una relación entre ellas, por lo que sugiere que existe una relación más compleja que no se puede apreciar mediante una inspección visual.

La diferencia que se encuentra entre los dos métodos es que las distribuciones de las clases de susceptibilidad del primer método se encuentran más dispersas a comparación del segundo método en la que se puede observar más homogeneidad entre las clases.

Figura 20

Susceptibilidad a movimientos en masa del área 1 y 2 mediante el primer método

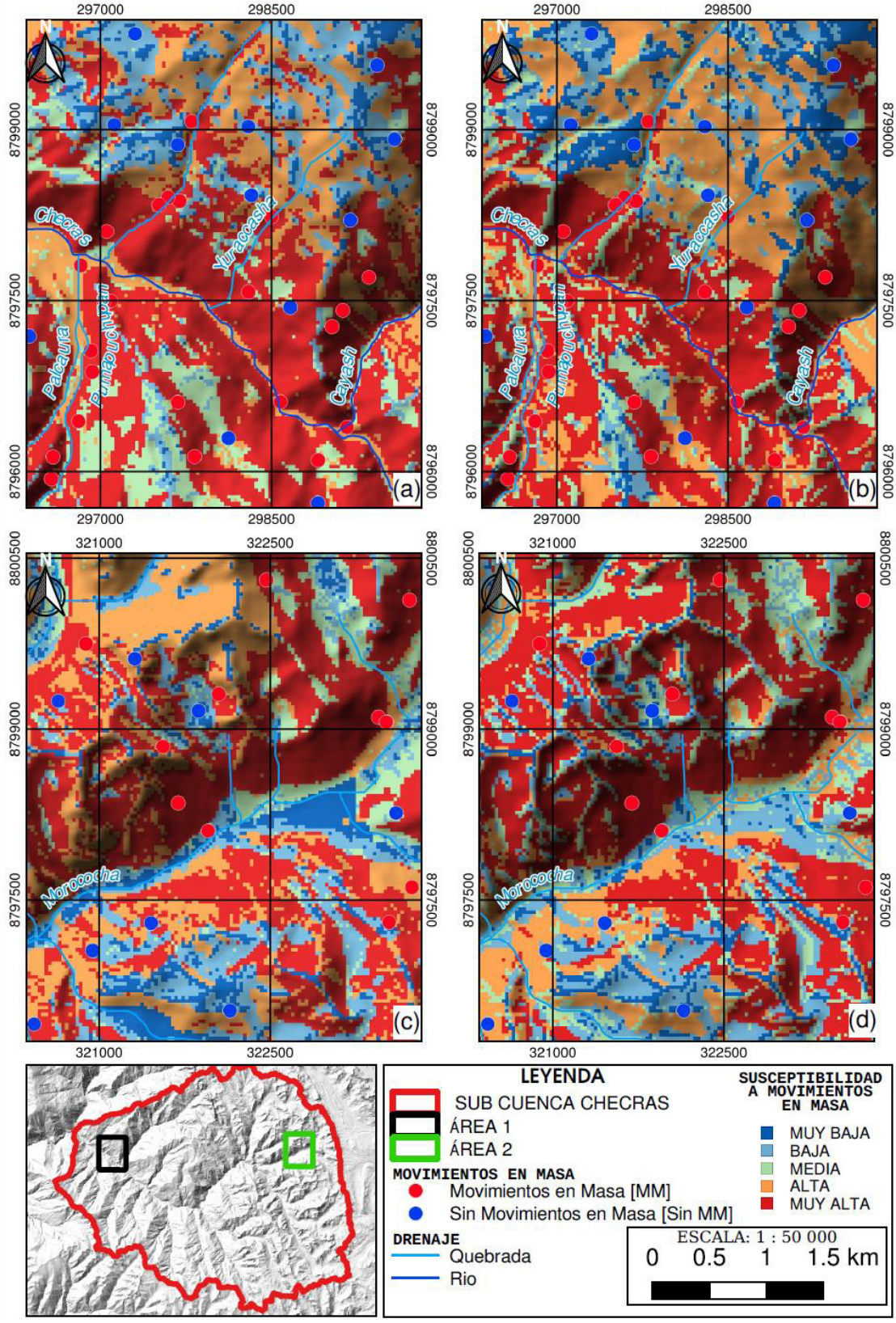
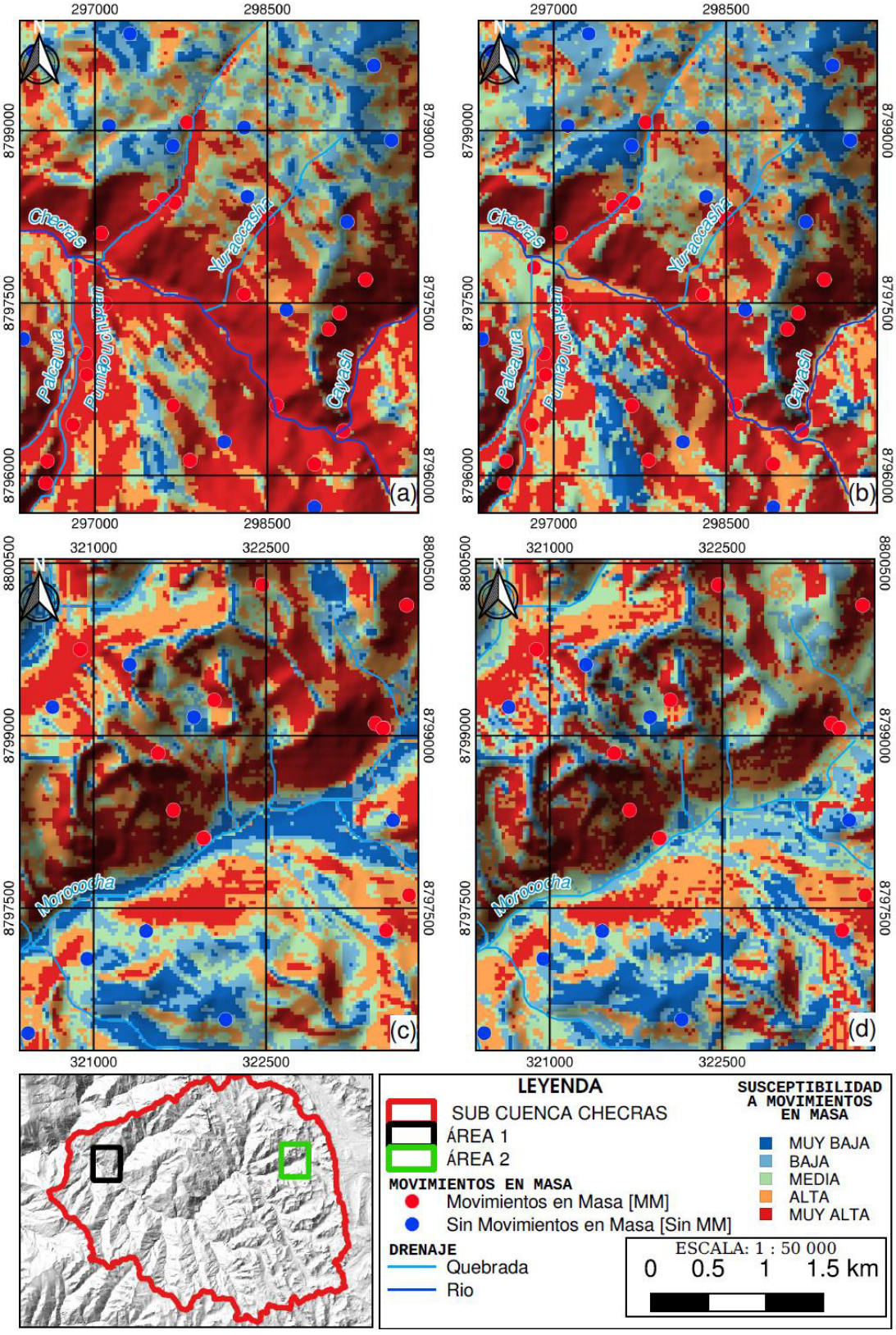


Figura 21

Susceptibilidad a movimientos en masa del área 1 y 2 mediante el segundo método



3.2. Aportes más destacables a la institución

El proyecto desarrollado en el presente informe de suficiencia profesional ha sido una iniciativa de gran relevancia para el inicio de la implementación de nuevos métodos de estudio aplicados a las geociencias. A continuación, se detallarán los beneficios más notables y relevantes que ha proporcionado:

3.2.1. Contribuciones específicas del proyecto

- Desarrollo de un método de procedimiento en el lenguaje de programación Python para la determinación de la susceptibilidad de movimientos en masa, mediante el uso de un modelo de aprendizaje supervisado y el empleo de factores condicionantes.
- Automatización del proceso de preprocesamiento de los factores condicionantes, para la obtención de insumos para los modelos.
- Determinación e implementación de modelos de aprendizaje automático óptimos para su aplicación en la obtención de la susceptibilidad a movimientos en masa.

3.2.2. Contribución en actividades posteriores

- Fortalecimiento en método para la generación de información geoespacial para la implementación de datos de susceptibilidad de movimientos en masa en geoportales, como en la plataforma de fotogrametría y teledección.

IV. CONCLUSIONES

Los movimientos en masa son desastres originados por fenómenos naturales que se debería tener muy en cuenta debido al peligro que significa en muchos lugares de nuestro país y el mundo. Su estudio y la obtención de información, como la estimación espacial de la susceptibilidad de este fenómeno hace que estemos más preparados en un posible desastre ocurrido por este fenómeno.

La aplicación de los modelos de aprendizaje automático para la estimación espacial de la susceptibilidad a movimientos en masa en la sub cuenca Checras mediante el uso de los factores condicionantes nos proporciona resultados satisfactorios para posibles usos como herramienta para la gestión de peligros asociados a estos fenómenos, sin embargo, es posible superar los resultados mediante la consideración de modelos más complejos, otros factores no usados y recolección de datos más precisos.

Los modelos de predicción entrenados con datos obtenidos del pre procesamiento mediante los dos métodos considerados, obtienen resultados ligeramente distintos los cuales se pueden contrastar mediante las métricas de evaluación de cada modelo. En el caso del primer método, el modelo SVM obtiene un AUC de 0.88, mientras que el modelo RF un AUC de 0.90, por esta razón, se considera que el modelo RF tiene un mejor rendimiento que el modelo SVM. En el caso del segundo método, el modelo SVM obtiene un AUC de 0.899 y el modelo RF un valor de 0.908, por lo que, también, en este método el modelo RF es el modelo con mejor rendimiento que el modelo SVM. De los dos métodos los modelos RF son lo que obtienen los valores de AUC más altos, sin embargo, el modelo RF del segundo método, el que fue pre procesado mediante *frequency ratio* y análisis de componente principales, obtiene un valor ligeramente superior al del primer método, en consecuencia, un mejor rendimiento en los resultados de la susceptibilidad a movimientos en masa.

La importancia de los nueve factores condicionantes considerados varía en cada uno de los modelos de cada uno de los métodos empleados. Estos factores tienen contribuciones, ya sea negativa o positiva en la determinación de los movimientos en masa. En el primer método el factor pendiente es uno de los principales para ambos modelos, seguido de la curvatura plana. El tercer y cuarto factor es la geomorfología y la litología para el modelo SVM y la inversa para el modelo RF. El quinto, séptimo y octavo factor son la curvatura de perfil, distancia a fallas y densidad de drenaje respectivamente, para ambos modelos. El sexto y noveno factor son el TWI y SAVI – NDWI para el modelo RF e inversa posición de los mismos factores para el modelo SVM debido a que los valores de TWI son negativos y su aporte en el modelo es mayormente a la predicción de la ausencia de movimientos en masa.

Una vez obtenido la susceptibilidad a movimientos en masa en la sub cuenca Checras se continuó a calcular la distribución mediante el clasificador por cortes naturales (Jenks) en la cual el modelo SVM del primer método obtiene 31.62 % del área total de muy alta susceptibilidad, 27.92 % alta, 6.57 % media, 20.16 % baja y 13.73 % muy baja. La distribución del modelo RF del primer método es de 29.67 % de muy alta susceptibilidad, 22.56 % alta, 19.58 % media, 17.72 % baja y 10.46 % muy baja. Para el segundo método, la distribución del modelo SVM es de 23.21 % de susceptibilidad muy alta, 23.39 % alta, 19.54 % media, 17.72 % baja y 16.14 % muy baja. Para el modelo SVM es de 22.37 % de susceptibilidad muy alta, 19.97 % alta, 20.59 % media, 18.98 % baja y 18.09 % muy baja.

V. RECOMENDACIONES

Teniendo en cuenta la importancia de la investigación se formula algunas recomendaciones, para los interesados, en el tema con el propósito de mejorar los resultados obtenidos.

- Realizar una recolección rigurosa tanto en gabinete como en campo de los datos de movimientos en masa en la totalidad de la sub cuenca con el fin de mejorar el entrenamiento de los modelos de aprendizaje automático.
- Experimentar otros tipos de pre procesamiento de los factores condicionantes como el método pesos de evidencia, que ayuden al entrenamiento de los modelos aplicados.
- Considera otros factores condicionantes en función a las características del área de estudio y a la relación con los movimientos en masa, como el tipo de suelo, el Índice de Diferencia Normalizada edificada (NDBI), Índice de Posición Topográfica (TPI), elevación, rugosidad del terreno, etc.
- Considerar, también, los factores desencadenantes como la precipitación, sismicidad, etc.
- Considerar modelos más complejos, en caso que sea necesario, con las que se puedan mejorar los resultados, como las redes neuronales artificiales.

VI. REFERENCIAS

Arabameri, A., Saha, S., Roy, J., Chen, W., Blaschke, T., y Tien Bui, D. (2020). Landslide susceptibility evaluation and management using different machine learning methods in the Gallicash river watershed, Iran. *Remote Sensing*, *12*(3), 475. https://doi.org/10.3390/rs12030475

Bakharia, A. (2016, febrero 1). *Visualising top features in linear SVM with scikit learn and matplotlib*. Medium. https://aneesha.medium.com/visualising-top-features-in-linear-svm-with-scikit-learn-and-matplotlib-3454ab18a14d

Chang, Z., Du, Z., Zhang, F., Huang, F., Chen, J., Li, W., y Guo, Z. (2020). Landslide susceptibility prediction based on remote sensing images and GIS: Comparisons of supervised and unsupervised machine learning models. *Remote Sensing*, *12*(3), 502. https://doi.org/10.3390/rs12030502

Cobbing, E. (1973). *Geología de los cuadrángulos de Barranca, Ámbar, Oyón, Huacho, Huaral y Canta (hojas 22-h, 22-i, 22-j, 23-h, 23-i, 23-j) - [Boletín A 26]*. Servicio de Geología y Minería.

Dangeti, P. (2017). *Statistics for Machine Learning*. Packt Publishing Ltd.

Fenner, M. (2019). *Machine learning with python for everyone*. Addison Wesley.

Fidel, L., y Zavala, B. (2006). Susceptibilidad a los movimientos en masa en la cuenca de la quebrada Hualanga. Pataz, La Libertad. *XIII Congreso Peruano de Geología*, 119–122. https://repositorio.ingemmet.gob.pe/handle/20.500.12544/452

Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2a ed.). O'Reilly Media.

Instituto Geológico, Minero y Metalúrgico. (2019). *Evaluación de peligros geológicos por movimientos en masa de las localidades de Cucho y Nueva Rinconada del centro poblado de Huanchayllo*. (A6936). Instituto Geológico, Minero y Metalúrgico - INGEMMET.

James, G., Witten, D., Hastie, T., y Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2a ed.). Springer.

Lee, S. (2014). *Geological Application of Geographic Information System*. Science Technology Network.

Raschka, S., y Mirjalili, V. (2019). *Python machine learning: Machine learning and deep learning with python, scikit-learn, and TensorFlow 2, 3rd edition* (3a ed.). Packt Publishing.

Vílchez Mata, M. S., y Medina Allcca, L. (2008). *Susceptibilidad a los movimientos en masa en las áreas de Chachapoyas y Luya - Amazonas - Perú: aplicación del método bivariante*. https://repositorio.ingemmet.gob.pe/handle/20.500.12544/3435

Turkey). *Environmental Earth Sciences*, *65*(7), 2161–2178. https://doi.org/10.1007/s12665-011-1196-4

Zhou, C., Yin, K., Cao, Y., Ahmed, B., Li, Y., Catani, F., y Pourghasemi, H. R. (2018). Landslide susceptibility modeling applying machine learning methods: A case study from Longju in the Three Gorges Reservoir area, China. *Computers & geosciences*, *112*, 23–37. https://doi.org/10.1016/j.cageo.2017.11.019

VII. ANEXOS

Anexo A. Litología

SIMB.	UNIDAD	LITOLOGÍA
Qh-al	Depósito aluvial	Acumulación de grava, arena, limo y arcilla con clastos subangulosos a angulosos de diferente composición.
PN-c	Grupo Calipuy	Andesitas, dacitas y riolitas de color gris pardo, marrón, morado, en bancos gruesos. Conglomerados y lutitas marrón rojizos.
Ki-f	Formación Farrat	Areniscas blancas, areniscas y limolitas rojizas, microconglomerados con clastos de cuarcitas.
Ki-chi	Formación Chimú	Areniscas cuarzosas blancas, limoarcillitas grises y niveles de carbón.
Ki-ca	Formación Carhuaz	Areniscas gris verdosas, lutitas negras y limolitas marrones.
Ki-oy	Formación Oyón	Areniscas, capas de carbón, restos de plantas.
Ki-s	Formación Santa	Areniscas, cuarcitas, lutitas, niveles de carbón
Ki-ph	Formación Pariahuanca	Calizas arenosas, areniscas calcáreas
Ki-chu	Formación Chúlec	Calizas arenosas, areniscas calcáreas en capas medianas, coloraciones parduscas a beige.
Ks-ce	Formación Celendín	Calizas en capas medianas, calizas nodulares, margas y areniscas calcáreas
Ks-j	Formación Jumasha	Calizas micríticas grises y calizas nodulares
Q-gl	Depósito glaciar	Depósitos morrénicos, bloques angulosos rellenados con arcillas, limos y arenas.
KP-dia	Diabasa	Diabasa
Ki-pt	Formación Pariatambo	Lutitas grises a negras, calizas bituminosas nodulares
KP-pcz	Pórfido cuarcífero	Pórfido cuarcífero

Anexo B. *Clasificación de la distancia a las fallas*

Distancia a las fallas	Clase
0 – 200	1
200 – 400	2
400 – 800	3
> 800	4

Anexo C. *Geomorfología*

SIMB.	SUB UNIDAD
Ti	Terraza indiferenciada
RM-ri	Montaña en roca intrusiva
RCE-rs	Colina estructural en roca sedimentaria
RM-rs	Montaña en roca sedimentaria
V-d	Vertiente coluvial de detritos
RCL-rv	Colina y lomada en roca volcánica
V-cd	Vertiente o piedemonte coluvio-deluvial
Mo	Morrenas
RCL-ri	Colina y lomada en roca intrusiva
RC-ri	Colina en roca intrusiva
RMCE-rs	Montañas y colinas estructurales en roca sedimentaria
RME-rs	Montaña estructural en roca sedimentaria
Vll-gl	Valle glaciar
Bo	Bofedales
RM-rv	Montaña en roca volcánica
RCL-rs	Colina y lomada en roca sedimentaria
RM-rvs	Montaña en roca volcano-sedimentaria
P-at	Vertiente o piedemonte aluvio-torrencial

Anexo D. *Rango de la densidad de drenaje*

Rango de la Densidad de drenaje	Densidad
0 – 0.22 km/km2	Baja
0.22 – 0.53 km/km2	Media baja
0.53 – 0.86 km/km2	Media
0.86 – 1.27 km/km2	Media alta
> 1.27 km/km2	Alta

Anexo E. *Rango de pendientes*

Rango de pendiente	Superficie topográfica
0°-1°	Terreno llano a algo inclinado
1°-5°	Terreno inclinado con pendiente suave
5°-15°	Pendiente moderada
15°-25°	Pendiente fuerte
25°-45°	Pendiente muy fuerte o escarpada
Mayor a 45°	Pendiente muy escarpada

Anexo F. *Rango del TWI*

TWI	POTENCIAL DE ACUMULACIÓN DE AGUA
<=5.01	Muy bajo
5.01-6.42	Bajo
6.42-8.27	Medio
8.27-11.96	Alto
>11.96	Muy Alto

Anexo G. *Rangos de la curvatura de perfil*

Rango de la Curvatura de perfil	Curvatura
<= -0.0025	Convexa
-0.0025-0.0025	Linear
> 0.0025	Cóncava

Anexo H. *Rangos de la curvatura plana*

Rango de la Curvatura de plana	Curvatura
<= -0.008	Cóncava
-0.008-0.008	Linear
> 0.008	Convexa

Anexo I. *Código del flujo de trabajo de los modelos de aprendizaje automático*

CÓDIGO - FLUJO DE TRABAJO DE LOS MODELOS DE MACHINE LEARNING

Sección A

En esta sección se:

1. Importar librerías
2. Leer archivos raster de los factores y movimientos en masa
3. Asignar nombres y rangos a las variables
4. Obtener estadística de las variables

In [1]:
```
# Importar as librerias
import os
import gdal
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import copy
```

In [2]:
```
# Asignar la dirección de los archivos
fn = '/home/hansenbg/Prueba_ML/Avance/RECOPILADO/RASTER/'
```

In [3]:
```
# Asignar una dirección de trabajo
os.chdir(fn)
os.getcwd()
```

Out[3]:
```
'/home/hansenbg/Prueba_ML/Avance/RECOPILADO/RASTER'
```

In [4]:
```
# Llamar los factores y los MM en archivo raster
bands_clip = [band for band in os.listdir(fn) if band[-4:]=='.tif']
bands_clip
```

Out[4]:
```
['DISTANCIA_VIAS_CLIP.tif',
 'TWI_CLIP.tif',
 'CURVATURA_PLANA_CLIP.tif',
 'GEOLOGIA_CLIP.tif',
 'PENDIENTE_CLIP.tif',
 'MM_RF_m2.tif',
 'MM_CLIP.tif',
 'MM_RF_m1.tif',
 'DEM_CLIP.tif',
 'SAVI_NDWI_CLIP.tif',
 'DISTANCIA_FALLAS_CLIP.tif',
 'CURVATURA_PERFIL_CLIP.tif',
 'MM_SVM_m1.tif',
 'DENSIDAD_DRENAJE_CLIP.tif',
 'MM_SVM_m2.tif',
 'GEOMORFOLOGIA_CLIP.tif']
```

In [5]:
```
# Leer los datos raster
Geomorfologia = 'GEOMORFOLOGIA_CLIP.tif'
SAVI = 'SAVI_NDWI_CLIP.tif'
Distancia_Fallas = 'DISTANCIA_FALLAS_CLIP.tif'
Densidad_drenaje = 'DENSIDAD_DRENAJE_CLIP.tif'
Litologia = 'GEOLOGIA_CLIP.tif'
Pendiente = 'PENDIENTE_CLIP.tif'
Curvatura_perfil = 'CURVATURA_PERFIL_CLIP.tif'
TWI = 'TWI_CLIP.tif'
Curvatura_plana = 'CURVATURA_PLANA_CLIP.tif'
MM = 'MM_CLIP.tif'

Geomorfologia_2 = gdal.Open(Geomorfologia)
SAVI_2 = gdal.Open(SAVI)
Distancia_Fallas_2 = gdal.Open(Distancia_Fallas)
Densidad_drenaje_2 = gdal.Open(Densidad_drenaje)
Litologia_2 = gdal.Open(Litologia)
Pendiente_2 = gdal.Open(Pendiente)
Curvatura_perfil_2 = gdal.Open(Curvatura_perfil)
TWI_2 = gdal.Open(TWI)
Curvatura_plana_2 = gdal.Open(Curvatura_plana)
MM_2 = gdal.Open(MM)

# Extraer los valores
Geomorfologia_3 = Geomorfologia_2.GetRasterBand(1).ReadAsArray().flatten()
SAVI_3 = SAVI_2.GetRasterBand(1).ReadAsArray().flatten()
Distancia_Fallas_3 = Distancia_Fallas_2.GetRasterBand(1).ReadAsArray().flatten()
Densidad_drenaje_3 = Densidad_drenaje_2.GetRasterBand(1).ReadAsArray().flatten()
Litologia_3 = Litologia_2.GetRasterBand(1).ReadAsArray().flatten()
Pendiente_3 = Pendiente_2.GetRasterBand(1).ReadAsArray().flatten()
Curvatura_perfil_3 = Curvatura_perfil_2.GetRasterBand(1).ReadAsArray().flatten()
TWI_3 = TWI_2.GetRasterBand(1).ReadAsArray().flatten()
Curvatura_plana_3 = Curvatura_plana_2.GetRasterBand(1).ReadAsArray().flatten()
MM_3 = MM_2.GetRasterBand(1).ReadAsArray().flatten()
```

In [6]:
```
# Obtenerlas propiedades de un raster
driver = Litologia_2.GetDriver()
col = Litologia_2.RasterXSize
rows = Litologia_2.RasterYSize
nelm = col*rows
print(rows,col,nelm)
```

```
1067 1275 1360425
```

In [7]:
```
# Eliminar los pixeles nulos (NaN) para obtener solo los pixeles de la cuenca

NanValues = np.where(Pendiente_3 == -99999)[0]

DATA = np.stack((Geomorfologia_3,SAVI_3,Distancia_Fallas_3,Densidad_drenaje_3,Litologia_
                 Pendiente_3,Curvatura_perfil_3,Curvatura_plana_3,TWI_3,MM_3),axis=1)

cP        = np.arange(0,nelm)
cPP       = np.delete(cP, NanValues, axis=0)

XX = np.delete(DATA, NanValues, axis=0)
```

In [8]:
```
print (DATA.shape)
print (XX.shape)
```

```
(1360425, 10)
(912878, 10)
```

In [9]:
```
# Crear el dataframe de la colección de datos
columnas = ['Geomorfologia','SAVI_NDWI', 'Distancia fallas', 'Densidad drenaje','Litolog
            'Pendiente', 'Curvatura de perfil', 'Curvatura plana', 'TWI', 'MM']
df_XX = pd.DataFrame (XX, columns = columnas)
df_XX.head()
```

Out[9]:

Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM
0	1.0	2.0	313.209198	0.0	5.0	21.935677	0.013199	0.061512	3.571980	-10.0
1	1.0	2.0	342.052643	0.0	5.0	34.218250	0.000914	-0.005603	4.001869	-10.0
2	1.0	2.0	371.079498	0.0	5.0	31.744429	-0.002203	-0.005814	4.338459	-10.0
3	1.0	2.0	400.249939	0.0	5.0	29.112917	-0.002491	-0.017185	5.071407	-10.0

In [13]:
```
# Hacer una copia del dataframe
df_XX_2 = copy.copy(df_XX)
df_XX_2.head()
```

Out[13]:

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM
0	1.0	2.0	313.209198	0.0	5.0	21.935677	0.013199	0.061512	3.571980	-10.0
1	1.0	2.0	342.052643	0.0	5.0	34.218250	0.000914	-0.005603	4.001869	-10.0
2	1.0	2.0	371.079498	0.0	5.0	31.744429	-0.002203	-0.005814	4.338459	-10.0
3	1.0	2.0	400.249939	0.0	5.0	29.112917	-0.002491	-0.017185	5.071407	-10.0
4	1.0	2.0	429.534637	0.0	5.0	24.607536	-0.000443	-0.017513	4.553838	-10.0

In [14]:
```
# Asignar los nombres de las observaciones de las variables cualitativas, cuantitativas

##### Variables cualitativas#####
# Asignar nombre a la Geomorfología
morfo_num = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,31]
morfo = ['RCE-rs','RCL-ri','RCL-rs','RCL-rv','RMCE-rs','RM-ri','RM-rs','RM-rv','RM-rvs',
         'Vll-gl','V-d','P-at','V-cd','Bo','Bo']

for i,j in zip(morfo_num,morfo):
    df_XX_2['Geomorfologia'][df_XX_2['Geomorfologia'] == i] = j

#Asignar nombres a SAVI - NDWI
#1-Agua,2-Suelo desnudo,3-Vegetación rala,4-Vegetación media,5-Vegetación vigorosa
SAVI_num = [1,2,3,4,5,6]
SAVI = ['Agua', 'Suelo desnudo','Vegetación dispersa','Vegetación moderada','Vegetación

for i,j in zip(SAVI_num,SAVI):
    df_XX_2['SAVI_NDWI'][df_XX_2['SAVI_NDWI']==i] = j

# Categorizar las distancias a las fallas
bins = [-1,200,400,800,1000,10000000]
label = ['0-200 m','200-400 m','400-800 m','800-1000 m','>1000 m']
df_XX_2['Distancia fallas'] = pd.cut(df_XX_2['Distancia fallas'], bins, labels=label)

# Categorizar la densidad de drenaje
# Los rangos de categorización de la Densidad de Drenaje se obtuvo mendiante el método j
# La reclasificación de Natural break[0.0,0.22236,0.52753,0.86298,1.26562,2.37828]
bins = [-1,0.22,0.53,0.86,1.27,3]
label = ['Baja', 'Media baja', 'Media', 'Media alta', 'Alta']
df_XX_2['Densidad drenaje'] = pd.cut(df_XX_2['Densidad drenaje'], bins, labels = label)

# Asigna nombre a la LITOLOGIA
lito_num = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
lito = ['Qh-al','PN-c','Ki-f','Ki-chi','Ki-ca','Ki-oy','Ki-s','Ki-chu','Ki-ph','Ks-ce','
        'KP-dia','Ki-pt','KP-pcz']

for i,j in zip(lito_num,lito):
    df_XX_2['Litologia'][df_XX_2['Litologia'] == i] = j

##### Variables cuantitativas#####

# Categorizar la pendiente
bins = [-1, 1, 5, 15, 25, 45, 90]
label = ['Muy baja', 'Baja', 'Media', 'Fuerte', 'Muy fuerte', 'Abrupta']

df_XX_2['Pendiente'] = pd.cut(df_XX_2['Pendiente'], bins, labels=label)

# Categorizar la Curvatura de Perfil
bins = [-1,-0.0025,0.0025,1]
label  = ['Convexa', 'Linear', 'Concava']

df_XX_2['Curvatura de perfil'] = pd.cut(df_XX_2['Curvatura de perfil'],bins,labels=label

# Categorizar la Curvatura de Plana
bins = [-25,-0.008,0.008,25]
label  = ['Concava', 'Linear', 'Convexa']

df_XX_2['Curvatura plana'] = pd.cut(df_XX_2['Curvatura plana'],bins,labels=label)

# Categorizar el TWI por el método natural break (Jenks) [1.98,5.01,6.42,8.27,11.96,22.9
bins = [1.98,5.01,6.42,8.27,11.96,22.95]
label  = ['Muy alto', 'Alto', 'Medio', 'Bajo', 'Muy bajo']

df_XX_2['TWI'] = pd.cut(df_XX_2['TWI'],bins,labels=label)

##### Movimiento en Masa #####

# Categorizar el MM
df_XX_2['MM'][df_XX_2['MM']== -10] = 'Data a predecir'
df_XX_2['MM'][df_XX_2['MM']== 1] = 'MM'
df_XX_2['MM'][df_XX_2['MM']== 0] = 'Sin MM'

df_XX_2.head()
```

```
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:1
8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:3
8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:6
9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
```

Out[14]:

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM
0	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Fuerte	Concava	Convexa	Muy alto	Data a predecir
1	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir
2	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir
3	RCE-rs	Suelo desnudo	400-800 m	Baja	Ki-ca	Muy fuerte	Linear	Concava	Alto	Data a predecir
4	RCE-rs	Suelo desnudo	400-800 m	Baja	Ki-ca	Fuerte	Linear	Concava	Muy alto	Data a predecir

In [15]:
```
# Estadística de las variables

# Obtener la información del dataframe [Nombre, Valores no NaN y el tipo de dato de las
df_XX_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 912878 entries, 0 to 912877
Data columns (total 10 columns):
 #   Column               Non-Null Count   Dtype   
---  ------               --------------   -----   
 0   Geomorfologia        912878 non-null  object  
 1   SAVI_NDWI            912878 non-null  object  
 2   Distancia fallas     912878 non-null  category
 3   Densidad drenaje     912878 non-null  category
 4   Litologia            912878 non-null  object  
 5   Pendiente            912878 non-null  category
 6   Curvatura de perfil  912878 non-null  category
 7   Curvatura plana      912878 non-null  category
 8   TWI                  912878 non-null  category
 9   MM                   912878 non-null  object  
dtypes: category(6), object(4)
memory usage: 33.1+ MB
```

In [16]:
```
# Descripción de las var. cuantitativas [Cantidad, media, desviación estandard, mínimo,
df_XX.iloc[:,5:9].describe()
```

Out[16]:

	Pendiente	Curvatura de perfil	Curvatura plana	TWI
count	912878.000000	912878.000000	912878.000000	912878.000000
mean	27.078819	-0.000144	0.000219	6.031451
std	11.238954	0.003824	0.071339	1.905293
min	0.000000	-0.047504	-20.824295	1.981781
25%	19.311764	-0.001984	-0.006332	4.791482
50%	27.827928	-0.000267	-0.000157	5.673806
75%	34.893112	0.001492	0.006106	6.798536

In [17]:
```
# Descripción de las variables cualitativas [Cantidad, Valores únicos, Moda, Frecuencia
df_XX_2.iloc[:,0:5].describe(include=['object','category'])
```

Out[17]:

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia
count	912878	912878	912878	912878	912878
unique	17	5	5	5	15
top	RM-rs	Suelo desnudo	>1000 m	Media	PN-c
freq	357463	620991	325532	239655	228869

Sección B

En esta sección se:

1. Obtener los valores de Frequency Ratio para todos los factores
2. Exportar estos valores a un excel

Preprocesamiento de las variables cualitativas y cuantitativas

Frequency Ratio

FR = %pixeles MM en cada factor / %pixeles totales de cada fator

%pixeles MM en cada factor = #Pix. de MM de cada clase del factor / #MM totales * 100

%de pixeles totales de cada fator = #Pix. del factor / #Pix totales * 100

In [20]:
```
# Hacer una copia de las variables cualitativas para obtener la proporción de frecuencia
df_XX_3 = copy.copy(df_XX_2)
df_XX_3.head()
```

Out[20]:

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM
0	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Fuerte	Concava	Convexa	Muy alto	Data a predecir
1	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir
2	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir
3	RCE-rs	Suelo desnudo	400-800 m	Baja	Ki-ca	Muy fuerte	Linear	Concava	Alto	Data a predecir
4	RCE-rs	Suelo desnudo	400-800 m	Baja	Ki-ca	Fuerte	Linear	Concava	Muy alto	Data a predecir

In [21]:
```
# Crear más columnas en base a las seleccionadas en col para determinar el número de pi

# PL (Pixeles de landslide)
col = df_XX_3.iloc[:,0:9].columns
for i in col:
    df_XX_3[i+'_PL'] = -99999

# Cuenta los números de pixeles con MM en cada clase de cada factor
col = df_XX_3.iloc[:,0:9].columns
for j in col:
    SAV = df_XX_3[j].unique()
    for i in SAV:
        df_XX_3[j+'_PL'][df_XX_3[j]==i] = df_XX_3[j][(df_XX_3[j] == i) & (df_XX_3['MM']

# Crear más columnas en base a las seleccionadas en col para determinar el
# número de pixeles de cada tipo de observación

# PT (Pixeles totales)
col = df_XX_3.iloc[:,0:9].columns
for i in col:
    df_XX_3[i+'_PT'] = -99999

## Cantidad de Pixeles de cada clase de cada factor
col = df_XX_3.iloc[:,0:9].columns
for j in col:
    SAV = df_XX_3[j].unique()
    for i in SAV:
        df_XX_3[j+'_PT'][df_XX_3[j]==i] = df_XX_3[j][(df_XX_3[j] == i)].count()

df_XX_3.head()
```

```
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:1
3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  del sys.path[0]
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:2
8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
```

Out[21]:

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM	...
0	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Fuerte	Concava	Convexa	Muy alto	Data a predecir	...
1	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir	...
2	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir	...
3	RCE-rs	Suelo desnudo	400-800 m	Baja	Ki-ca	Muy fuerte	Linear	Concava	Alto	Data a predecir	...
4	RCE-rs	Suelo desnudo	400-800 m	Baja	Ki-ca	Fuerte	Linear	Concava	Muy alto	Data a predecir	...

5 rows × 28 columns

In [22]:
```
## Pasar a porcentaje los resultados anteriores

# Hacer una copia
df_XX_4 = copy.copy(df_XX_3)
df_XX_4.head()
```

Out[22]:

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM	...

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM	...
0	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Fuerte	Concava	Convexa	Muy alto	Data a predecir	...
1	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir	...
2	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir	...
3	RCE-rs	Suelo desnudo	400-800 m	Baja	Ki-ca	Muy fuerte	Linear	Concava	Alto	Data a predecir	...
4	RCE-rs	Suelo	400-800	Baja	Ki-ca	Fuerte	Linear	Concava	Muy	Data a	...

In [23]:
```
# Crear campos donde se calculará los porcentajes del PL
col = df_XX_4.iloc[:,10:19].columns
for i in col:
    df_XX_4[i+'_%'] = -99999

# Calcular los porcentajes del PL
col = df_XX_4.iloc[:,10:19].columns
for i in col:
    df_XX_4[i+'_%'] = (df_XX_4[i]*100)/df_XX_4['MM'][df_XX_4['MM']=='MM'].count()

# Crear campos donde se calculará los porcentajes del PT
col = df_XX_4.iloc[:,19:28].columns
for i in col:
    df_XX_4[i+'_%'] = -99999

# Calcular los porcentajes del PT
col = df_XX_4.iloc[:,19:28].columns
for i in col:
    df_XX_4[i+'_%'] = (df_XX_4[i]*100)/df_XX_4.shape[0]

df_XX_4.head()
```

Out[23]:

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM	...
0	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Fuerte	Concava	Convexa	Muy alto	Data a predecir	...
1	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir	...
2	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir	...
3	RCE-rs	Suelo desnudo	400-800 m	Baja	Ki-ca	Muy fuerte	Linear	Concava	Alto	Data a predecir	...
4	RCE-rs	Suelo desnudo	400-800 m	Baja	Ki-ca	Fuerte	Linear	Concava	Muy alto	Data a predecir	...

5 rows × 46 columns

In [24]:
```
## Proporción DE FRECUENCIA (Frequency Ratio) por cada factor

# Crear más columnas en base a las seleccionadas en col para determinar FR de cada clase
col = df_XX_4.iloc[:,0:9].columns
for i in col:
    df_XX_4[i+'_FR'] = -99999

# Determinar la Proporción DE FRECUENCIA por cada factor
percent_land = df_XX_4.iloc[:,28:37].columns

percent_tot = df_XX_4.iloc[:,37:46].columns

col_FR = df_XX_4.iloc[:,46:55].columns

for i,j,z in zip(percent_land,percent_tot,col_FR):
    df_XX_4[z] = df_XX_4[i]/df_XX_4[j]

df_XX_4.head()
```

Out[24]:

	Geomorfologia	SAVI_NDWI	Distancia fallas	Densidad drenaje	Litologia	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM	...
0	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Fuerte	Concava	Convexa	Muy alto	Data a predecir	...
1	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir	...
2	RCE-rs	Suelo desnudo	200-400 m	Baja	Ki-ca	Muy fuerte	Linear	Linear	Muy alto	Data a predecir	...
3	RCE-rs	Suelo desnudo	400-800 m	Baja	Ki-ca	Muy fuerte	Linear	Concava	Alto	Data a predecir	...
4	RCE-rs	Suelo desnudo	400-800 m	Baja	Ki-ca	Fuerte	Linear	Concava	Muy alto	Data a predecir	...

5 rows × 55 columns

In [25]:
```
# Seleccionar las columnas de ratio de frecuencia de los factores
df_fact_FR = copy.copy (df_XX_4[col_FR])
df_fact_FR.head()
```

Out[25]:

	Geomorfologia_FR	SAVI_NDWI_FR	Distancia fallas_FR	Densidad drenaje_FR	Litologia_FR	Pendiente_FR	Curvatura de perfil_FR	Curvatura plana_FR
0	0.751336	1.169439	1.035483	0.794988	0.718453	0.492795	0.667790	0.477333
1	0.751336	1.169439	1.035483	0.794988	0.718453	1.398849	1.009804	1.158623
2	0.751336	1.169439	1.035483	0.794988	0.718453	1.398849	1.009804	1.158623
3	0.751336	1.169439	1.003411	0.794988	0.718453	1.398849	1.009804	1.057796
4	0.751336	1.169439	1.003411	0.794988	0.718453	0.492795	1.009804	1.057796

In [24]:

```
# Formar cuadro de Frequency Ratio para exportar a Excel

Geomorfologia = df_XX_4[['Geomorfologia','Geomorfologia_PL','Geomorfologia_PT','Geomorfo
                         'Geomorfologia_PT_%','Geomorfologia_FR']]
Geomorfologia['Factor'] = 'Geomorfologia'

Geomorfologia.columns = ['Clases','#Pixeles MM','#Pixeles totales','%Pixeles MM','%Pixel
                         'Ratio de Frecuencia','Factor']

SAVI_NDWI = df_XX_4[['SAVI_NDWI','SAVI_NDWI_PL','SAVI_NDWI_PT','SAVI_NDWI_PL_%','SAVI_ND
                     'SAVI_NDWI_FR']]
SAVI_NDWI['Factor'] = 'SAVI_NDWI'

SAVI_NDWI.columns = ['Clases','#Pixeles MM','#Pixeles totales','%Pixeles MM','%Pixeles t
                     'Ratio de Frecuencia','Factor']

Distancia_fallas = df_XX_4[['Distancia fallas','Distancia fallas_PL','Distancia fallas_P
                            'Distancia fallas_PL_%','Distancia fallas_PT_%','Distancia f
Distancia_fallas['Factor'] = 'Distancia fallas'

Distancia_fallas.columns = ['Clases','#Pixeles MM','#Pixeles totales','%Pixeles MM','%Pi
                            'Ratio de Frecuencia','Factor']

Densidad_drenaje = df_XX_4[['Densidad drenaje','Densidad drenaje_PL','Densidad drenaje_P
                            'Densidad drenaje_PL_%','Densidad drenaje_PT_%','Densidad dr
Densidad_drenaje['Factor'] = 'Densidad drenaje'

Densidad_drenaje.columns = ['Clases','#Pixeles MM','#Pixeles totales','%Pixeles MM','%Pi
                            'Ratio de Frecuencia','Factor']

Litologia = df_XX_4[['Litologia','Litologia_PL','Litologia_PT','Litologia_PL_%','Litolog
                     'Litologia_FR']]
Litologia['Factor'] = 'Litologia'

Litologia.columns = ['Clases','#Pixeles MM','#Pixeles totales','%Pixeles MM','%Pixeles t
                     'Ratio de Frecuencia','Factor']

Pendiente = df_XX_4[['Pendiente','Pendiente_PL','Pendiente_PT','Pendiente_PL_%','Pendien
                     'Pendiente_FR']]
Pendiente['Factor'] = 'Pendiente'

Pendiente.columns = ['Clases','#Pixeles MM','#Pixeles totales','%Pixeles MM','%Pixeles t
                     'Ratio de Frecuencia','Factor']

Curvatura_perfil = df_XX_4[['Curvatura de perfil','Curvatura de perfil_PL','Curvatura de
                            'Curvatura de perfil_PL_%','Curvatura de perfil_PT_%','Curva
Curvatura_perfil['Factor'] = 'Curvatura de perfil'

Curvatura_perfil.columns = ['Clases','#Pixeles MM','#Pixeles totales','%Pixeles MM','%Pi
                            'Ratio de Frecuencia','Factor']

Curvatura_plana = df_XX_4[['Curvatura plana','Curvatura plana_PL','Curvatura plana_PT','
                           'Curvatura plana_PT_%','Curvatura plana_FR']]
Curvatura_plana['Factor'] = 'Curvatura plana'

Curvatura_plana.columns = ['Clases','#Pixeles MM','#Pixeles totales','%Pixeles MM','%Pix
                           'Ratio de Frecuencia','Factor']

TWI = df_XX_4[['TWI','TWI_PL','TWI_PT','TWI_PL_%','TWI_PT_%','TWI_FR']]
TWI['Factor'] = 'TWI'

TWI.columns = ['Clases','#Pixeles MM','#Pixeles totales','%Pixeles MM','%Pixeles total',
               'Ratio de Frecuencia','Factor']

Factores = pd.concat([Geomorfologia, SAVI_NDWI, Distancia_fallas, Densidad_drenaje, Litol
                      Pendiente, Curvatura_perfil, Curvatura_plana, TWI])

table = pd.pivot_table(Factores, values=['#Pixeles MM','#Pixeles totales','%Pixeles MM',
                                         'Ratio de Frecuencia'], index=['Factor', 'Clase
                       aggfunc=np.mean, fill_value=0)

# Exporta a excel
table.to_excel('Frequency_Ratio.xlsx')

table
```

```
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  after removing the cwd from sys.path.
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:1
0: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:1
6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  app.launch_new_instance()
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:2
2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:2
8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:3
4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:4
0: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:4
6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
/home/hansenbg/anaconda3/envs/py_37/lib/python3.7/site-packages/ipykernel_launcher.py:5
3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
```

Out[24]:

Factor	Clases	#Pixeles MM	#Pixeles totales	%Pixeles MM	%Pixeles total	Ratio de Frecuencia
Curvatura de perfil	Concava	77	147423	10.784314	16.149255	0.667790
	Convexa	174	179238	24.369748	19.634387	1.241177
	Linear	463	586217	64.845938	64.216357	1.009804
Curvatura plana	Concava	69	184817	9.663866	20.245531	0.477333

Factor	Clases	#Pixeles MM	#Pixeles totales	%Pixeles MM	%Pixeles total	Ratio de Frecuencia
	Convexa	155	187346	21.708683	20.522567	1.057796
...	...	...	...	...	...	...
TWI	Alto	333	337331	46.638655	36.952473	1.262125
	Bajo	30	69072	4.201681	7.566400	0.555308
	Medio	149	207166	20.868347	22.693722	0.919565
	Muy alto	193	283719	27.030812	31.079619	0.869728
	Muy bajo	9	15590	1.260504	1.707786	0.738093

Sección C

En esta sección se:

1. Obtener los componentes principales de las variables cuantitativas

Preprocesamiento de las variables cuantitativas

Análisis de los componentes principales (PCA)

In [26]:
```
# Hacer una copia de las variables cuantitativas para obtener los PCAs

df_XX_5 = copy.copy(pd.concat([df_XX[['Pendiente', 'Curvatura de perfil', 'Curvatura pla
                                      'TWI']],df_XX_2.iloc[:,9]],axis=1))
df_XX_5.head()
```

Out[26]:

	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM
0	21.935677	0.013199	0.061512	3.571980	Data a predecir
1	34.218250	0.000914	-0.005603	4.001869	Data a predecir
2	31.744429	-0.002203	-0.005814	4.338459	Data a predecir
3	29.112917	-0.002491	-0.017185	5.071407	Data a predecir
4	24.607536	-0.000443	-0.017513	4.553838	Data a predecir

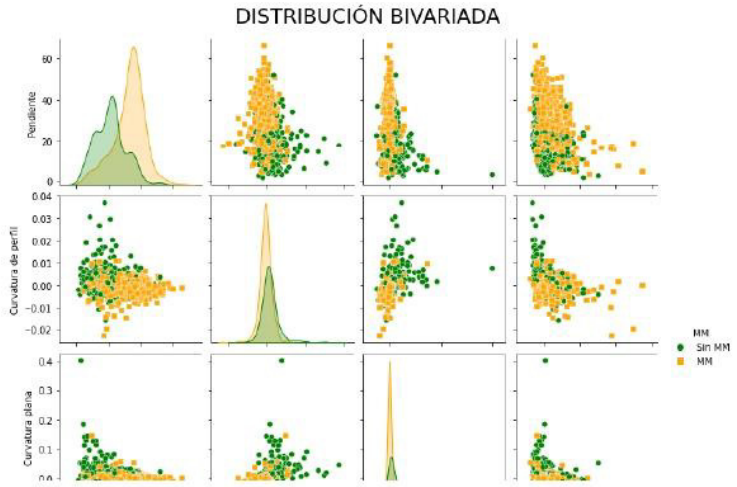
In [27]:
```
# Seleccionar los pixeles con datos de MM y no MM
df_XX_6 = df_XX_5[(df_XX_5['MM'] == 'MM') | (df_XX_5['MM'] == 'Sin MM')]
df_XX_6.head()
```

Out[27]:

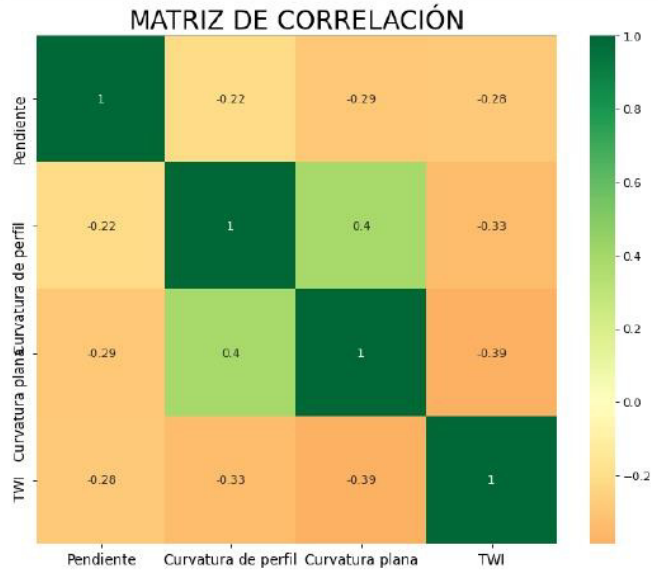
	Pendiente	Curvatura de perfil	Curvatura plana	TWI	MM
1466	3.792046	0.002974	0.046697	5.980696	Sin MM
1775	12.081735	0.003934	0.016504	5.031220	Sin MM
2536	14.266019	-0.010904	-0.008703	6.219070	Sin MM
4674	22.612226	-0.009284	-0.024716	6.655383	Sin MM
5272	22.590208	-0.000233	0.004247	5.478796	MM

In [30]:
```
# Distribución bivariada de las variables numéricas
plt.figure(figsize=(10,8), dpi= 80)
sns.pairplot(df_XX_6,
             hue = 'MM',
             markers=["o", "s"], palette=['g', 'orange']
             )
plt.suptitle('DISTRIBUCIÓN BIVARIADA', y=1.03, fontsize=22)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```

```
<Figure size 800x640 with 0 Axes>
```



In [31]:
```
# Matriz de correlación
# Plot
plt.figure(figsize=(10,8), dpi= 80)
sns.heatmap(df_XX_6.iloc[:,:4].corr(), xticklabels=df_XX_6.iloc[:,:4].corr().columns,
            yticklabels=df_XX_6.iloc[:,:4].corr().columns, cmap='RdYlGn', center=0, annc

# Decorations
plt.title('MATRIZ DE CORRELACIÓN', fontsize=22)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```



In [32]:
```
# Estandarizar las variables (TOTAL)

#from sklearn.preprocessing import StandardScaler
#x = StandardScaler().fit_transform(df_XX_5.iloc[:,:4])
#x

from sklearn.preprocessing import MinMaxScaler
x = MinMaxScaler().fit_transform(df_XX_5.iloc[:,:4])
x
```

Out[32]:
```
array([[0.29043418, 0.48345578, 0.49837446, 0.07585125],
       [0.45305872, 0.38561752, 0.49677297, 0.09635659],
       [0.42030466, 0.36078802, 0.49676794, 0.11241166],
       ...,
       [0.43037438, 0.40323222, 0.49710035, 0.09726159],
       [0.3439986 , 0.40119326, 0.49672133, 0.10665891],
       [0.27564788, 0.49762505, 0.49693733, 0.09868649]], dtype=float32)
```

In [33]:
```
# Asignar los nombres de las columnas
standardS_x = pd.DataFrame(x,columns=['Pendiente', 'Curvatura de perfil', 'Curvatura pla
standardS_x.head()
```

Out[33]:

	Pendiente	Curvatura de perfil	Curvatura plana	TWI
0	0.290434	0.483456	0.498374	0.075851
1	0.453059	0.385618	0.496773	0.096357
2	0.420305	0.360788	0.496768	0.112412
3	0.385463	0.358495	0.496497	0.147373
4	0.325810	0.374805	0.496489	0.122685

In [34]:
```
# PCA
from sklearn.decomposition import PCA
pca = PCA(n_components=4)
PCA_TOTAL = pca.fit_transform(x)
print("The explained variance por 1 PC is: ", np.sum(pca.explained_variance_ratio_))
```

```
The explained variance por 1 PC is:  0.9999999999999999
```

In [35]:
```
# Varianza total explicada
# Verificar que parte de la variabilidad se incorpora en cada componente principal
pca.explained_variance_ratio_
```

Out[35]:
```
array([8.04070970e-01, 1.71647834e-01, 2.41911433e-02, 9.00525207e-05])
```

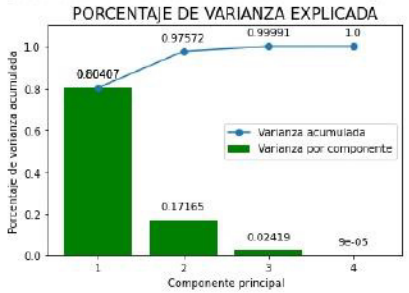
In [36]:
```
PCA = pd.DataFrame(PCA_TOTAL, columns = ['PC1','PC2','PC3','PC4'])
PCA_concat = pd.concat([PCA,df_XX_5.iloc[:,4]],axis=1)
PCA_concat.head()
```

Out[36]:

	PC1	PC2	PC3	PC4	MM
0	-0.015005	-0.151846	0.079541	0.000741	Data a predecir
1	0.125030	-0.052458	-0.003717	-0.000378	Data a predecir
2	0.087964	-0.045546	-0.026996	-0.000299	Data a predecir
3	0.042345	-0.027409	-0.025034	-0.000487	Data a predecir
4	-0.002596	-0.075855	-0.016832	-0.000677	Data a predecir

In [37]:
```
# Porcentaje de varianza explicada acumulada

prop_varianza_acum = pca.explained_variance_ratio_.cumsum()
print('----------------------------------------')
print('Porcentaje de varianza explicada acumulada')
print('----------------------------------------')
print(prop_varianza_acum)

fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(6, 4))

ax.plot(
    np.arange(len(PCA.columns)) + 1,
    prop_varianza_acum, marker = 'o', label = 'Varianza acumulada')

ax.bar(
    x      = np.arange(pca.n_components_) + 1,
    height = pca.explained_variance_ratio_, color='green', label = 'Varianza por compone

for x, y in zip(np.arange(len(PCA.columns)) + 1, prop_varianza_acum):
    label = round(y, 5)
    ax.annotate(
        label,
        (x,y),
        textcoords="offset points",
        xytext=(0,10),
        ha='center')

for x, y in zip(np.arange(len(PCA.columns)) + 1, pca.explained_variance_ratio_):
    label = round(y, 5)
    ax.annotate(
        label,
        (x,y),
        textcoords="offset points",
        xytext=(0,10),
        ha='center')

ax.set_ylim(0, 1.1)
ax.set_xticks(np.arange(pca.n_components_) + 1)
ax.set_title('PORCENTAJE DE VARIANZA EXPLICADA',fontsize=15)
ax.set_xlabel('Componente principal')
ax.set_ylabel('Porcentaje de varianza acumulada')
ax.legend(loc = 'center right')
plt.show()
```

```
----------------------------------------
Porcentaje de varianza explicada acumulada
----------------------------------------
[0.80407097 0.9757188  0.99990995 1.        ]
```



In [38]:
```
# Seleccionar los pixeles con datos de MM y no MM para la gráfica en 3d
PCA_concat_MM = PCA_concat[(PCA_concat['MM'] == 'MM') | (PCA_concat['MM'] == 'Sin MM')]
PCA_concat_MM.head()
```

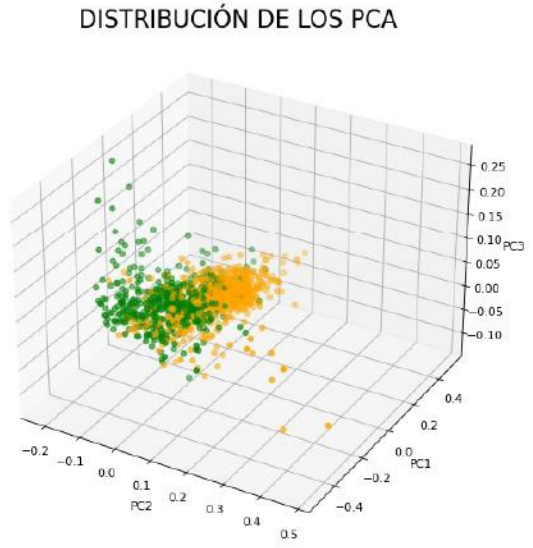
Out[38]:

	PC1	PC2	PC3	PC4	MM
1466	-0.282325	-0.126258	0.006429	0.000730	Sin MM

	PC1	PC2	PC3	PC4	MM
1775	-0.163479	-0.126060	0.012059	-0.000047	Sin MM
2536	-0.161163	-0.041590	-0.092203	-0.000130	Sin MM
4674	-0.067380	0.017585	-0.069516	-0.000369	Sin MM

In [39]:

```
fig = plt.figure(figsize=(10,8), dpi= 80)

ax = fig.add_subplot(111, projection='3d')

x = PCA_concat_MM['PC2']
y = PCA_concat_MM['PC1']
z = PCA_concat_MM['PC3']

col = PCA_concat_MM['MM'].map({'Sin MM':'g', 'MM':'orange'})

ax.scatter(x,y,z, c=col)

ax.set_xlabel('PC2')
ax.set_ylabel('PC1')
ax.set_zlabel('PC3')
#ax.legend()
plt.title('DISTRIBUCIÓN DE LOS PCA', fontsize = 20)
plt.show()
```



Sección D

En esta sección se:

1. Aplicar los modelos del primer método
2. Separar los datos en entrenamiento y testeo
3. Entrenar los modelos
4. Evaluar los modelos mediante las métricas
5. Reporte de los resultados
6. Exportar en mapa los resultados

1° MÉTODO: Frequency Ratio

In [36]:
```
# Concatenar los resultados de Proporción de Frecuencia y las variables cuantitativas
df_metodo_1 = pd.concat([df_fact_FR,df_XX.iloc[:,9]],axis=1)
df_metodo_1
```

Out[36]:

	Geomorfologia_FR	SAVI_NDWI_FR	Distancia fallas_FR	Densidad drenaje_FR	Litologia_FR	Pendiente_FR	Curvatura de perfil_FR	Curva plana
0	0.751336	1.169439	1.035483	0.794988	0.718453	0.492795	0.667790	0.47
1	0.751336	1.169439	1.035483	0.794988	0.718453	1.398849	1.009804	1.15
2	0.751336	1.169439	1.035483	0.794988	0.718453	1.398849	1.009804	1.15
3	0.751336	1.169439	1.003411	0.794988	0.718453	1.398849	1.009804	1.05
4	0.751336	1.169439	1.003411	0.794988	0.718453	0.492795	1.009804	1.05
...	...	...	...	...	...	...	...	
912873	0.751336	1.169439	1.003411	0.794988	0.793261	1.398849	1.009804	0.47
912874	0.751336	1.169439	1.003411	0.794988	0.793261	1.398849	0.667790	0.47
912875	0.751336	1.169439	1.003411	0.794988	0.793261	1.398849	0.667790	0.47
912876	0.751336	1.169439	1.003411	0.794988	0.793261	1.398849	0.667790	1.15
912877	0.751336	1.169439	1.003411	0.794988	0.793261	0.492795	0.667790	1.15

912878 rows × 10 columns

In [38]:
```
# Seleccionar los pixeles con datos de MM y no MM para la aplicación de los modelos
df_metodo_1_mod = df_metodo_1[(df_metodo_1['MM'] == 1) | (df_metodo_1['MM'] == 0)]
df_metodo_1_mod
```

Out[38]:

	Geomorfologia_FR	SAVI_NDWI_FR	Distancia fallas_FR	Densidad drenaje_FR	Litologia_FR	Pendiente_FR	Curvatura de perfil_FR	Curva plana
1466	0.751336	0.719040	1.003411	0.704513	0.718453	0.233449	0.667790	0.47
1775	0.751336	1.169439	1.025088	0.704513	1.012493	0.397984	0.667790	0.47
2536	0.751336	0.646057	1.003411	0.949616	0.631725	0.397984	1.241177	1.05
4674	0.751336	1.169439	1.025088	0.794988	1.012493	0.492795	1.241177	1.05
5272	0.751336	1.169439	1.025088	0.704513	1.012493	0.492795	1.009804	1.15
...	...	...	...	...	...	...	...	
907946	0.866085	1.169439	1.003411	0.794988	0.793261	1.398849	1.009804	1.15
908447	0.751336	1.169439	1.035483	0.794988	0.793261	0.397984	0.667790	1.05
909113	1.085533	1.169439	1.003411	0.794988	0.793261	1.398849	1.241177	1.15
909697	0.751336	1.169439	1.035483	0.949616	1.699032	1.398849	1.241177	1.15

Geomorfologia_FR SAVI_NDWI_FR Distancia fallas_FR Densidad drenaje_FR Litologia_FR Pendiente_FR Curvatura de perfil_FR Curva plana

In [39]:
```
# Seleccionar las variables en:
# X: Variables independientes
# Y: Variables dependientes

X_m1 = df_metodo_1_mod.iloc[:,:9]
Y_m1 = df_metodo_1_mod.iloc[:,9]
```

In [40]:
```
## Dividir la información en train y test [Para el entrenamiento y testeo]
from sklearn.model_selection import train_test_split
X_train_m1, X_test_m1, Y_train_m1, Y_test_m1 = train_test_split (X_m1, Y_m1,
                                                                 test_size = 0.3,
                                                                 random_state = 123)

print (f'X_train_m1 : {X_train_m1.shape}')
print (f'Y_train_m1 : {Y_train_m1.shape}')
print (f'X_test_m1 : {X_test_m1.shape}')
print (f'Y_test_m1 : {Y_test_m1.shape}')
```

```
X_train_m1 : (851, 9)
Y_train_m1 : (851,)
X_test_m1 : (366, 9)
Y_test_m1 : (366,)
```

1° método: Máquina de Soporte Vectorial

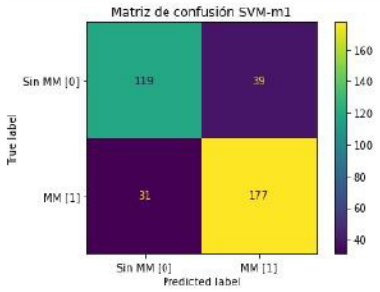
* Entrenamiento

In [41]:
```
# Entrenar el modelo
from sklearn.svm import SVC
cls_svm_m1 = SVC(kernel ='linear', random_state = 0, probability = True)
cls_svm_m1.fit(X_train_m1, Y_train_m1)
cls_svm_m1
```

Out[41]:
```
SVC(kernel='linear', probability=True, random_state=0)
```

* Evaluación

In [42]:
```
# Evaluación del modelo SVM

# Matriz de confusión
from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(cls_svm_m1,X_test_m1,Y_test_m1,
                      values_format = 'd', display_labels =['Sin MM [0]', 'MM [1]'])
plt.title('Matriz de confusión SVM-m1')
plt.show()
```



In [43]:
```
# Accuracy
from sklearn.metrics import accuracy_score
Y_train_m1_pred_svm = cls_svm_m1.predict(X_train_m1)
Y_test_m1_pred_svm = cls_svm_m1.predict(X_test_m1)
print ('Accuracy de entrenamiento del modelo SVM-m1: ' +
       str(accuracy_score(Y_train_m1, Y_train_m1_pred_svm)))
print ('Accuracy de prueba del modelo SVM-m1: ' +
       str(accuracy_score(Y_test_m1, Y_test_m1_pred_svm)))
```

```
Accuracy de entrenamiento del modelo SVM-m1: 0.7943595769682726
Accuracy de prueba del modelo SVM-m1: 0.8087431693989071
```

1° método: Bosques Aleatorios

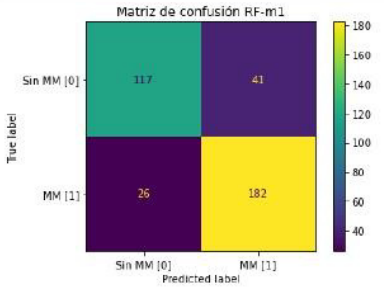
* Entrenamiento

In [44]:
```
# Entrenamiento del modelo
from sklearn.ensemble import RandomForestClassifier
cls_rf_m1 = RandomForestClassifier(n_estimators = 50, random_state = 0, min_samples_leaf
cls_rf_m1.fit(X_train_m1, Y_train_m1)
cls_rf_m1
```

Out[44]:
```
RandomForestClassifier(max_depth=5, min_samples_leaf=8, n_estimators=50,
                       random_state=0)
```

* Validación

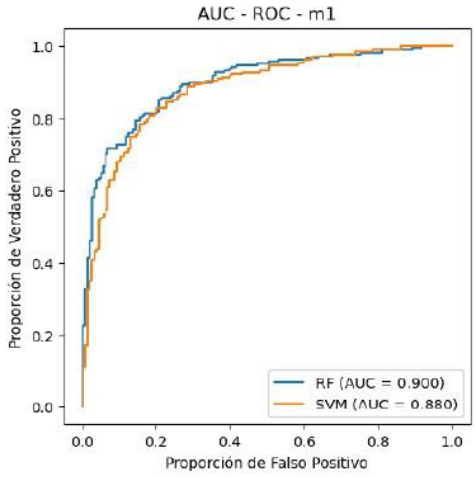
In [45]:
```
# Evaluación del modelo RF

# Matriz de confusión
from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(cls_rf_m1,X_test_m1,Y_test_m1,
                      values_format = 'd', display_labels =['Sin MM [0]', 'MM [1]'])
plt.title('Matriz de confusión RF-m1')
plt.show()
```



In [46]:
```
# Accuracy
from sklearn.metrics import accuracy_score
Y_train_m1_pred_rf = cls_rf_m1.predict(X_train_m1)
Y_test_m1_pred_rf = cls_rf_m1.predict(X_test_m1)
print ('Accuracy de entrenamiento del modelo RF-m1 ' +
       str(accuracy_score(Y_train_m1, Y_train_m1_pred_rf)))
print ('Accuracy de prueba del modelo RF-m1 ' +
       str(accuracy_score(Y_test_m1, Y_test_m1_pred_rf)))
```

```
Accuracy de entrenamiento del modelo RF-m1 0.836662749706228
Accuracy de prueba del modelo RF-m1 0.8169398907103825
```

In [47]:
```
# Curva ROC de los modelos SVM y RF

from sklearn.metrics import roc_curve, auc

#Obtener las probabilidades de cada modelo
Y_test_m1_prob_svm = cls_svm_m1.predict_proba(X_test_m1)[:,1]
Y_test_m1_prob_rf = cls_rf_m1.predict_proba(X_test_m1)[:,1]

# Random Forest
rf_fpr_m1, rf_tpr_m1, threshold_m1 = roc_curve (Y_test_m1, Y_test_m1_prob_rf)
auc_rf_m1 = auc (rf_fpr_m1, rf_tpr_m1)
# Suport Vector Machine
svm_fpr_m1, svm_tpr_m1, threshold_m1 = roc_curve (Y_test_m1, Y_test_m1_prob_svm)
auc_svm_m1 = auc (svm_fpr_m1, svm_tpr_m1)


plt.figure (figsize = (5,5), dpi = 100)
plt.plot (rf_fpr_m1, rf_tpr_m1, linestyle = '-', label = 'RF (AUC = %0.3f)'%auc_rf_m1)
plt.plot (svm_fpr_m1, svm_tpr_m1, linestyle = '-', label = 'SVM (AUC = %0.3f)'%auc_svm_m

plt.title ('AUC - ROC - m1')
plt.xlabel('Proporción de Falso Positivo')
plt.ylabel('Proporción de Verdadero Positivo')

plt.legend()
plt.show()
```



* Reporte

In [49]:
```
# Distribución de la susceptibilidad a MM
```

In [48]:
```
# Predecir la probabilidad de ocurrencia de MM de los datos totales [En toda la sub cuer
Y_total_m1_prob_svm= cls_svm_m1.predict_proba(df_metodo_1.iloc[:,0:9])[:,1]
Y_total_m1_prob_rf= cls_rf_m1.predict_proba(df_metodo_1.iloc[:,0:9])[:,1]

# Añadir a un dataframe
MM_m1 = pd.DataFrame(Y_total_m1_prob_svm,columns=['SVM'])
MM_m1['RF'] = pd.DataFrame(Y_total_m1_prob_rf, columns=['RF'])
print (MM_m1.shape)
MM_m1.head()
```

```
(912878, 2)
```

Out[48]:

	SVM	RF
0	0.173547	0.183635
1	0.804211	0.712647
2	0.804211	0.712647
3	0.760912	0.774442
4	0.256207	0.417266

In [51]:

```
# Clasificar los MM según el método Jenks
import jenkspy

breaks_SVM_m1 = jenkspy.jenks_breaks(MM_m1['SVM'], nb_class = 5) # 5 Clases
print (breaks_SVM_m1)
print ('SVM Clasificado: '  + str(breaks_SVM_m1))

breaks_RF_m1 = jenkspy.jenks_breaks(MM_m1['RF'], nb_class = 5) # 5 Clases
print (breaks_RF_m1)
print ('RF Clasificado: ' + str(breaks_RF_m1))
```

```
SVM Clasificado: [0.020980455611003687, 0.21950538624322644, 0.3346250651001956, 0.58229
40938200291, 0.8071777162994962, 0.981608096927668]
RF Clasificado: [0.07311402678915967, 0.2690592904818168, 0.438989663646632, 0.624036596
4052572, 0.796333693054829, 0.9749237727942999]
```

In [53]:

```
print ('SVM Clasificado: '  + str(breaks_SVM_m1))
# [0.020980455611003687, 0.21950538624322644, 0.3346250651001956, 0.5822940938200291, 0.
print ('RF Clasificado: ' + str(breaks_RF_m1))
# [0.07311402678915967, 0.2690592904818168, 0.438989663646632, 0.6240365964052572, 0.796
```

```
SVM Clasificado: [0.020980455611003687, 0.21950538624322644, 0.3346250651001956, 0.58229
40938200291, 0.8071777162994962, 0.981608096927668]
RF Clasificado: [0.07311402678915967, 0.2690592904818168, 0.438989663646632, 0.624036596
4052572, 0.796333693054829, 0.9749237727942999]
```

In [50]:

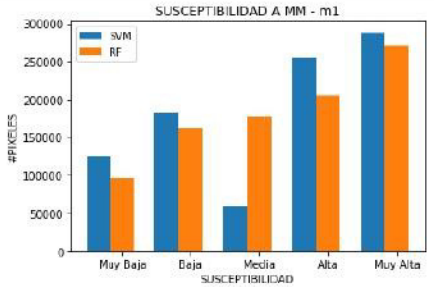
```
# Categorizar los MM del modelo SVM
bins = [0,0.22,0.33,0.58,0.81,1.0]
label = ['Muy Baja', 'Baja', 'Media', 'Alta', 'Muy Alta']
MM_m1['SVM'] = pd.cut(MM_m1['SVM'], bins, labels = label)

# Categorizar los MM del modelo RF
bins = [0,0.27,0.44,0.62,0.79,1.0]
label = ['Muy Baja', 'Baja', 'Media', 'Alta', 'Muy Alta']
MM_m1['RF'] = pd.cut(MM_m1['RF'], bins, labels = label)

MM_m1.head()
```

Out[50]:

	SVM	RF
0	Muy Baja	Muy Baja
1	Alta	Alta
2	Alta	Alta
3	Alta	Alta
4	Baja	Baja

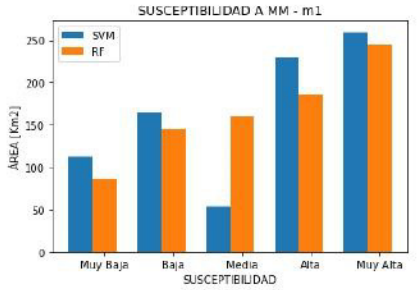
In [52]:

```
# Definir parámetros para la gráfica Distribución de la susceptibilidad a MM (En # Pixe
numero_grupos = len(MM_m1['SVM'].value_counts())  #Primero se debe realizar el agrupamie
indice_barras = np.arange(numero_grupos)
ancho_barras = 0.35
DATA_SVM = [MM_m1['SVM'].value_counts().sort_index()]
DATA_RF = [MM_m1['RF'].value_counts().sort_index()]

# Gráfica de la distribución de la susceptibilidad a MM
plt.bar(indice_barras, DATA_SVM[0],width=ancho_barras, label='SVM')
plt.bar(indice_barras+ancho_barras, DATA_RF[0], width=ancho_barras, label='RF')
plt.xticks(indice_barras+ancho_barras,('Muy Baja', 'Baja', 'Media', 'Alta', 'Muy Alta'))
plt.legend()
plt.ylabel('#PIXELES')
plt.xlabel('SUSCEPTIBILIDAD')
plt.title('SUSCEPTIBILIDAD A MM - m1')
plt.show()
```



In [69]:

```
# Definimos parámetros para la gráfica Distribución de la susceptibilidad a MM  (En Km2,
numero_grupos = len(MM_m1['SVM'].value_counts())  # Primero se debe realizar el agrupami
indice_barras = np.arange(numero_grupos)
ancho_barras = 0.35
DATA_SVM = [MM_m1['SVM'].value_counts().sort_index()*(900/1000000)]
DATA_RF = [MM_m1['RF'].value_counts().sort_index()*(900/1000000)]

# Gráfica de la distribución de la susceptibilidad a MM
plt.bar(indice_barras, DATA_SVM[0],width=ancho_barras, label='SVM')
plt.bar(indice_barras+ancho_barras, DATA_RF[0], width=ancho_barras, label='RF')
plt.xticks(indice_barras+ancho_barras,('Muy Baja', 'Baja', 'Media', 'Alta', 'Muy Alta'))
plt.legend()
plt.ylabel('ÁREA [Km2]')
plt.xlabel('SUSCEPTIBILIDAD')
plt.title('SUSCEPTIBILIDAD A MM - m1')
plt.show()
```



In [53]:

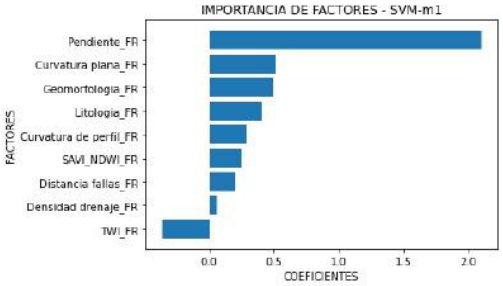
```
# Contribución de variables [Importancia de las variables]
```

In [87]:
```
# Importancia de las variables según el modelo SVM
IV_SVM_m1 = cls_svm_m1.coef_[0]
print (IV_SVM_m1)

# Importancia de las variables según el modelo RF
IV_RF_m1 = cls_rf_m1.feature_importances_
print (IV_RF_m1)
```

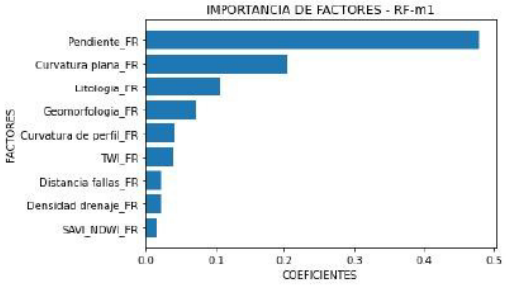
```
[ 0.49806007  0.24522016  0.19537434  0.05844282  0.40823181  2.10094426
  0.28958827  0.51507278 -0.36593347]
[0.07123836 0.01602112 0.02204599 0.02143752 0.10671958 0.47898534
 0.0411652  0.20386944 0.03851744]
```

In [88]:
```
# Extraer propiedades de uno de los modelos
numero_grupos = len(IV_RF_m1)
indice_barras = np.arange(numero_grupos)
ancho_barras = 0.8
fact_m1 = df_metodo_1.columns[:9]
```

In [89]:
```
# Graficar la importancia de SVM
imp_SVM_m1,names_SVM_m1 = zip(*sorted(zip(IV_SVM_m1,fact_m1))) # Se ordena de forma decr
plt.barh(indice_barras, imp_SVM_m1, label='SVM')

plt.yticks(indice_barras,names_SVM_m1)

plt.ylabel('FACTORES')
plt.xlabel('COEFICIENTES')
plt.title('IMPORTANCIA DE FACTORES - SVM-m1')
plt.show()
```



In [90]:
```
# Graficar la importancia de RF
imp_RF_m1,names_RF_m1 = zip(*sorted(zip(IV_RF_m1,fact_m1))) # Se ordena de forma decreci
plt.barh(indice_barras, imp_RF_m1, label='SVM')

plt.yticks(indice_barras,names_RF_m1)

plt.ylabel('FACTORES')
plt.xlabel('COEFICIENTES')
plt.title('IMPORTANCIA DE FACTORES - RF-m1')
plt.show()
```



Generación del mapa de susceptibilidad MM

In [91]:
```
# PREDICE VALORES SVM

## GUARDAR RESULTADO EN UN RASTER
SVM_RASTER = driver.Create(fn + "MM_SVM_m1" + ".tif", col, rows, 1, gdal.GDT_Float32)
# Write metadata
SVM_RASTER.SetGeoTransform(Litologia_2.GetGeoTransform())
SVM_RASTER.SetProjection(Litologia_2.GetProjection())

SVM_RASTERdataarray = np.zeros((rows,col)).flatten()

for i in range(cPP.shape[0]):
    SVM_RASTERdataarray[cPP[i]]=Y_total_m1_prob_svm[i]

for i in range(NanValues.shape[0]):
    SVM_RASTERdataarray[NanValues[i]]=-99999

SVM_RASTER.GetRasterBand(1).WriteArray(SVM_RASTERdataarray.reshape(rows,col))
SVM_RASTER.GetRasterBand(1).SetNoDataValue(-99999)
SVM_RASTER = None
del SVM_RASTER
```

In [67]:
```
# PREDICE VALORES RF

## GUARDAR RESULTADO EN UN RASTER
RF_RASTER = driver.Create(fn + "MM_RF_m1" + ".tif", col, rows, 1, gdal.GDT_Float32)
# Write metadata
RF_RASTER.SetGeoTransform(Litologia_2.GetGeoTransform())
RF_RASTER.SetProjection(Litologia_2.GetProjection())

RF_RASTERdataarray = np.zeros((rows,col)).flatten()

for i in range(cPP.shape[0]):
    RF_RASTERdataarray[cPP[i]]=Y_total_m1_prob_rf[i]

for i in range(NanValues.shape[0]):
    RF_RASTERdataarray[NanValues[i]]=-99999

RF_RASTER.GetRasterBand(1).WriteArray(RF_RASTERdataarray.reshape(rows,col))
RF_RASTER.GetRasterBand(1).SetNoDataValue(-99999)
RF_RASTER = None
del RF_RASTER
```

Sección E

En esta sección se:

1. Aplicar los modelos del segundo método
2. Separar los datos en entrenamiento y testeo
3. Entrenar los modelos
4. Evaluar los modelos mediante las métricas
5. Reporte de los resultados
6. Exportar en mapa los resultados

2° MÉTODO: PCA / Frequency Ratio

In [92]:
```
# Concatenar los resultados de Frequency Ratio y las variables cuantitativas
df_metodo_2 = pd.concat([df_fact_FR.iloc[:,0:5],PCA_concat.loc[:,['PC1','PC2','PC3']],df
df_metodo_2
```

Out[92]:

	Geomorfologia_FR	SAVI_NDWI_FR	Distancia fallas_FR	Densidad drenaje_FR	Litologia_FR	PC1	PC2	PC3
0	0.751336	1.169439	1.035483	0.794988	0.718453	-0.015005	-0.151846	0.079541
1	0.751336	1.169439	1.035483	0.794988	0.718453	0.125030	-0.052458	-0.003717
2	0.751336	1.169439	1.035483	0.794988	0.718453	0.087964	-0.045546	-0.026996
3	0.751336	1.169439	1.003411	0.794988	0.718453	0.042345	-0.027409	-0.025034
4	0.751336	1.169439	1.003411	0.794988	0.718453	-0.002596	-0.075855	-0.016832
...	...	...	...	...	...	...	...	...
912873	0.751336	1.169439	1.003411	0.794988	0.793261	0.053253	-0.069274	-0.000037
912874	0.751336	1.169439	1.003411	0.794988	0.793261	0.072572	-0.076967	0.028222
912875	0.751336	1.169439	1.003411	0.794988	0.793261	0.104080	-0.063630	0.012477
912876	0.751336	1.169439	1.003411	0.794988	0.793261	0.020885	-0.088218	0.007247
912877	0.751336	1.169439	1.003411	0.794988	0.793261	-0.037147	-0.139938	0.096644

912878 rows × 9 columns

In [93]:
```
# Seleccionar los pixeles con datos de MM y no MM para la aplicación de los modelos
df_metodo_2_mod = df_metodo_2[(df_metodo_2['MM'] == 1) | (df_metodo_2['MM'] == 0)]
df_metodo_2_mod
```

Out[93]:

	Geomorfologia_FR	SAVI_NDWI_FR	Distancia fallas_FR	Densidad drenaje_FR	Litologia_FR	PC1	PC2	PC3
1466	0.751336	0.719040	1.003411	0.704513	0.718453	-0.282325	-0.126258	0.006429
1775	0.751336	1.169439	1.025088	0.704513	1.012493	-0.163479	-0.126060	0.012059
2536	0.751336	0.646057	1.003411	0.949616	0.631725	-0.161163	-0.041590	-0.092203
4674	0.751336	1.169439	1.025088	0.794988	1.012493	-0.067380	0.017585	-0.069516
5272	0.751336	1.169439	1.025088	0.704513	1.012493	-0.044397	-0.046686	-0.008798
...	...	...	...	...	...	...	...	...
907946	0.866085	1.169439	1.003411	0.794988	0.793261	0.147533	0.000640	-0.001112
908447	0.751336	1.169439	1.035483	0.794988	0.793261	-0.205431	-0.020925	0.044501
909113	1.085533	1.169439	1.003411	0.794988	0.793261	0.131222	-0.026161	-0.050848
909697	0.751336	1.169439	1.035483	0.949616	1.699032	0.098547	-0.009526	-0.026835
912366	1.085533	1.169439	1.035483	0.794988	0.793261	0.130918	0.001462	-0.007530

1217 rows × 9 columns

In [94]:
```
# Seleccionar las variables en:

# X: Variables independientes
# Y: Variables dependientes

X_2 = df_metodo_2_mod.iloc[:,:8]
Y_2 = df_metodo_2_mod.iloc[:,8]
```

In [95]:
```
## Dividir la información en train y test
from sklearn.model_selection import train_test_split
X_train_m2, X_test_m2, Y_train_m2, Y_test_m2 = train_test_split (X_2, Y_2, test_size = 0

print (f'X_train_m2 : {X_train_m2.shape}')
print (f'Y_train_m2 : {Y_train_m2.shape}')
print (f'X_test_m2 : {X_test_m2.shape}')
print (f'Y_test_m2 : {Y_test_m2.shape}')
```

```
X_train_m2 : (851, 8)
Y_train_m2 : (851,)
X_test_m2 : (366, 8)
Y_test_m2 : (366,)
```

2° método: Máquina de Soporte Vectorial

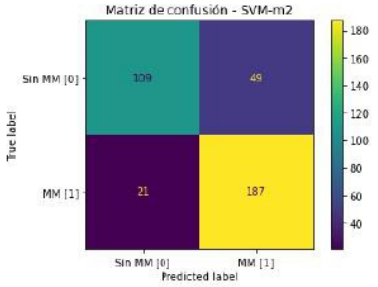
* Entrenamiento

In [96]:
```
# Entrenamiento del modelo
from sklearn.svm import SVC
cls_svm_m2 = SVC(kernel ='linear', random_state = 0, probability = True)
cls_svm_m2.fit(X_train_m2, Y_train_m2)
cls_svm_m2
```

Out[96]:
```
SVC(kernel='linear', probability=True, random_state=0)
```

* Validación

In [97]:
```
# Validación del modelo SVM

# Matriz de confusión
from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(cls_svm_m2,X_test_m2,Y_test_m2,
                      values_format = 'd', display_labels =['Sin MM [0]', 'MM [1]'])
plt.title('Matriz de confusión - SVM-m2')
plt.show()
```



In [98]:
```
# Accuracy
from sklearn.metrics import accuracy_score
Y_train_m2_pred_svm = cls_svm_m2.predict(X_train_m2)
Y_test_m2_pred_svm = cls_svm_m2.predict(X_test_m2)
print ('Accuracy de entrenamiento del modelo SVM ' + str(accuracy_score(Y_train_m2, Y_tr
print ('Accuracy de prueba del modelo SVM ' + str(accuracy_score(Y_test_m2, Y_test_m2_pr
```

```
Accuracy de entrenamiento del modelo SVM 0.7931844888366627
Accuracy de prueba del modelo SVM 0.8087431693989071
```

2° método: Bosques Aleatorios

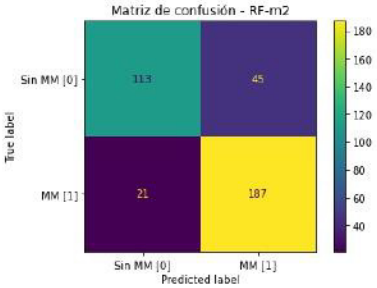
* Entrenamiento

In [99]:
```
# Entrenamiento del modelo
from sklearn.ensemble import RandomForestClassifier
cls_rf_m2 = RandomForestClassifier(n_estimators = 50, random_state = 0, min_samples_leaf
cls_rf_m2.fit(X_train_m2, Y_train_m2)
cls_rf_m2
```

Out[99]:
```
RandomForestClassifier(max_depth=5, min_samples_leaf=8, n_estimators=50,
                       random_state=0)
```

* Validación

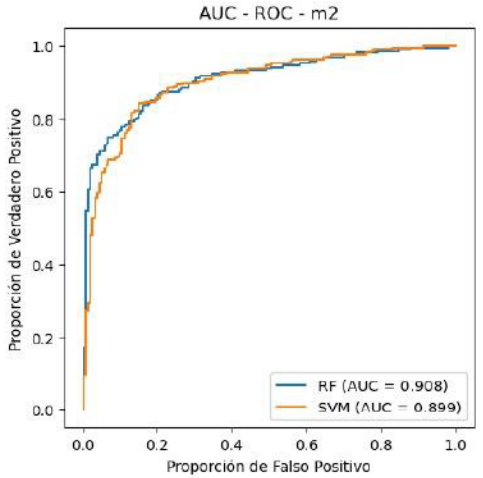
In [100…
```
# Validación del modelo RF

# Matriz de confusión
from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(cls_rf_m2,X_test_m2,Y_test_m2,
                      values_format = 'd', display_labels =['Sin MM [0]', 'MM [1]'])
plt.title('Matriz de confusión - RF-m2')
plt.show()
```



In [101…
```
# Accuracy
from sklearn.metrics import accuracy_score
# Primero se obtiene las predicciones del modelo
Y_train_m2_pred_rf = cls_rf_m2.predict(X_train_m2)
Y_test_m2_pred_rf = cls_rf_m2.predict(X_test_m2)
print ('Accuracy de entrenamiento del modelo RF-m2 ' + str(accuracy_score(Y_train_m2, Y_
print ('Accuracy de prueba del modelo RF-m2 ' + str(accuracy_score(Y_test_m2, Y_test_m2_
```

```
Accuracy de entrenamiento del modelo RF-m2 0.8472385428907168
Accuracy de prueba del modelo RF-m2 0.819672131147541
```

In [102…
```
# Curva ROC de los modelos SVM y RF

from sklearn.metrics import roc_curve, auc

#Obtener las probabilidades de cada modelo
Y_test_m2_prob_svm = cls_svm_m2.predict_proba(X_test_m2)[:,1]
Y_test_m2_prob_rf = cls_rf_m2.predict_proba(X_test_m2)[:,1]

# Random Forest
rf_fpr_m2, rf_tpr_m2, threshold_m2 = roc_curve (Y_test_m2, Y_test_m2_prob_rf)
auc_rf_m2 = auc (rf_fpr_m2, rf_tpr_m2)
# Suport Vector Machine
svm_fpr_m2, svm_tpr_m2, threshold_m2 = roc_curve (Y_test_m2, Y_test_m2_prob_svm)
auc_svm_m2 = auc (svm_fpr_m2, svm_tpr_m2)


plt.figure (figsize = (5,5), dpi = 100)
plt.plot (rf_fpr_m2, rf_tpr_m2, linestyle = '-', label = 'RF (AUC = %0.3f)'%auc_rf_m2)
plt.plot (svm_fpr_m2, svm_tpr_m2, linestyle = '-', label = 'SVM (AUC = %0.3f)'%auc_svm_m

plt.title ('AUC - ROC - m2')
plt.xlabel('Proporción de Falso Positivo')
plt.ylabel('Proporción de Verdadero Positivo')

plt.legend()
plt.show()
```



* Reporte

In [103…
```
# Distribución de la susceptibilidad a MM
```

In [104…
```
# Predecir la probabilidad de ocurrencia de MM de los datos totales [En toda la sub cuen
Y_total_m2_prob_svm= cls_svm_m2.predict_proba(df_metodo_2.iloc[:,0:8])[:,1]
Y_total_m2_prob_rf= cls_rf_m2.predict_proba(df_metodo_2.iloc[:,0:8])[:,1]

# Añadir a un dataframe
MM_m2 = pd.DataFrame(Y_total_m2_prob_svm,columns=['SVM'])
MM_m2['RF'] = pd.DataFrame(Y_total_m2_prob_rf, columns=['RF'])
print (MM_m2.shape)
MM_m2.head()
```

```
(912878, 2)
```

Out[104…

	SVM	RF
0	0.163785	0.191195
1	0.605738	0.454768
2	0.568349	0.497004
3	0.540324	0.621109
4	0.345537	0.260763

In [70]:

```
# Clasificar los MM según el método Jenks
import jenkspy

breaks_SVM_m2 = jenkspy.jenks_breaks(MM_m2['SVM'], nb_class = 5) # 5 Clases
print (breaks_SVM_m2)

breaks_RF_m2 = jenkspy.jenks_breaks(MM_m2['RF'], nb_class = 5) # 5 Clases
print (breaks_RF_m2)
```

In [82]:

```
print ('SVM Clasificado: '  + str(breaks_SVM_m2))
# [0.0034052362375595695, 0.24249070070954376, 0.4408464910807822, 0.6285979243486707, 0
print ('RF Clasificado: ' + str(breaks_RF_m2))
# [0.06918145391384302, 0.3286984839242988, 0.5047235221264047, 0.6577753720603023, 0.80
```

```
SVM Clasificado: [0.0034052362375595695, 0.24249070070954376, 0.4408464910807822, 0.6285
979243486707, 0.8072292340190769, 0.9999999682145781]
RF Clasificado: [0.06918145391384302, 0.3286984839242988, 0.5047235221264047, 0.65777537
20603023, 0.8069808337917599, 0.9717384991311663]
```

In [105…

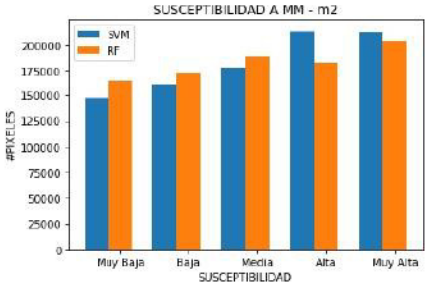
```
# Categorizar los MM del modelo SVM
bins = [0,0.24,0.44,0.63,0.81,1]
label = ['Muy Baja', 'Baja', 'Media', 'Alta', 'Muy Alta']
MM_m2['SVM'] = pd.cut(MM_m2['SVM'], bins, labels = label)

# Categorizar los MM del modelo RF
bins = [0,0.33,0.50,0.68,0.81,1]
label = ['Muy Baja', 'Baja', 'Media', 'Alta', 'Muy Alta']
MM_m2['RF'] = pd.cut(MM_m2['RF'], bins, labels = label)

MM_m2.head()
```

Out[105…

	SVM	RF
0	Muy Baja	Muy Baja
1	Media	Baja
2	Media	Baja
3	Media	Media
4	Baja	Muy Baja

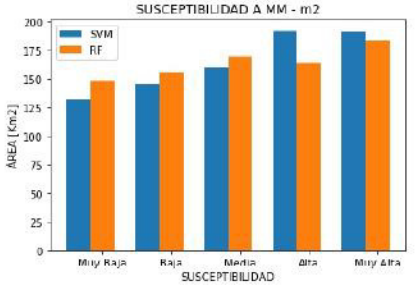
In [106…

```
# Definir parámetros para la gráfica Distribución de la susceptibilidad a MM
numero_grupos = len(MM_m2['SVM'].value_counts())  # Primero se debe realizar el agrupami
indice_barras = np.arange(numero_grupos)
ancho_barras = 0.35
DATA_SVM = [MM_m2['SVM'].value_counts().sort_index()]
DATA_RF = [MM_m2['RF'].value_counts().sort_index()]

# Gráfica de la distribución de la susceptibilidad a MM
plt.bar(indice_barras, DATA_SVM[0],width=ancho_barras, label='SVM')
plt.bar(indice_barras+ancho_barras, DATA_RF[0], width=ancho_barras, label='RF')
plt.xticks(indice_barras+ancho_barras,('Muy Baja', 'Baja', 'Media', 'Alta', 'Muy Alta'))
plt.legend()
plt.ylabel('#PIXELES')
plt.xlabel('SUSCEPTIBILIDAD')
plt.title('SUSCEPTIBILIDAD A MM - m2')
plt.show()
```



In [109…

```
# Definimos parámetros para la gráfica Distribución de la susceptibilidad a MM
numero_grupos = len(MM_m2['SVM'].value_counts())  # Primero se debe realizar el agrupami
indice_barras = np.arange(numero_grupos)
ancho_barras = 0.35
DATA_SVM = [MM_m2['SVM'].value_counts().sort_index()*(900/1000000)]
DATA_RF = [MM_m2['RF'].value_counts().sort_index()*(900/1000000)]

# Gráfica de la distribución de la susceptibilidad a MM
plt.bar(indice_barras, DATA_SVM[0],width=ancho_barras, label='SVM')
plt.bar(indice_barras+ancho_barras, DATA_RF[0], width=ancho_barras, label='RF')
plt.xticks(indice_barras+ancho_barras,('Muy Baja', 'Baja', 'Media', 'Alta', 'Muy Alta'))
plt.legend()
plt.ylabel('ÁREA [Km2]')
plt.xlabel('SUSCEPTIBILIDAD')
plt.title('SUSCEPTIBILIDAD A MM - m2')
plt.show()
```



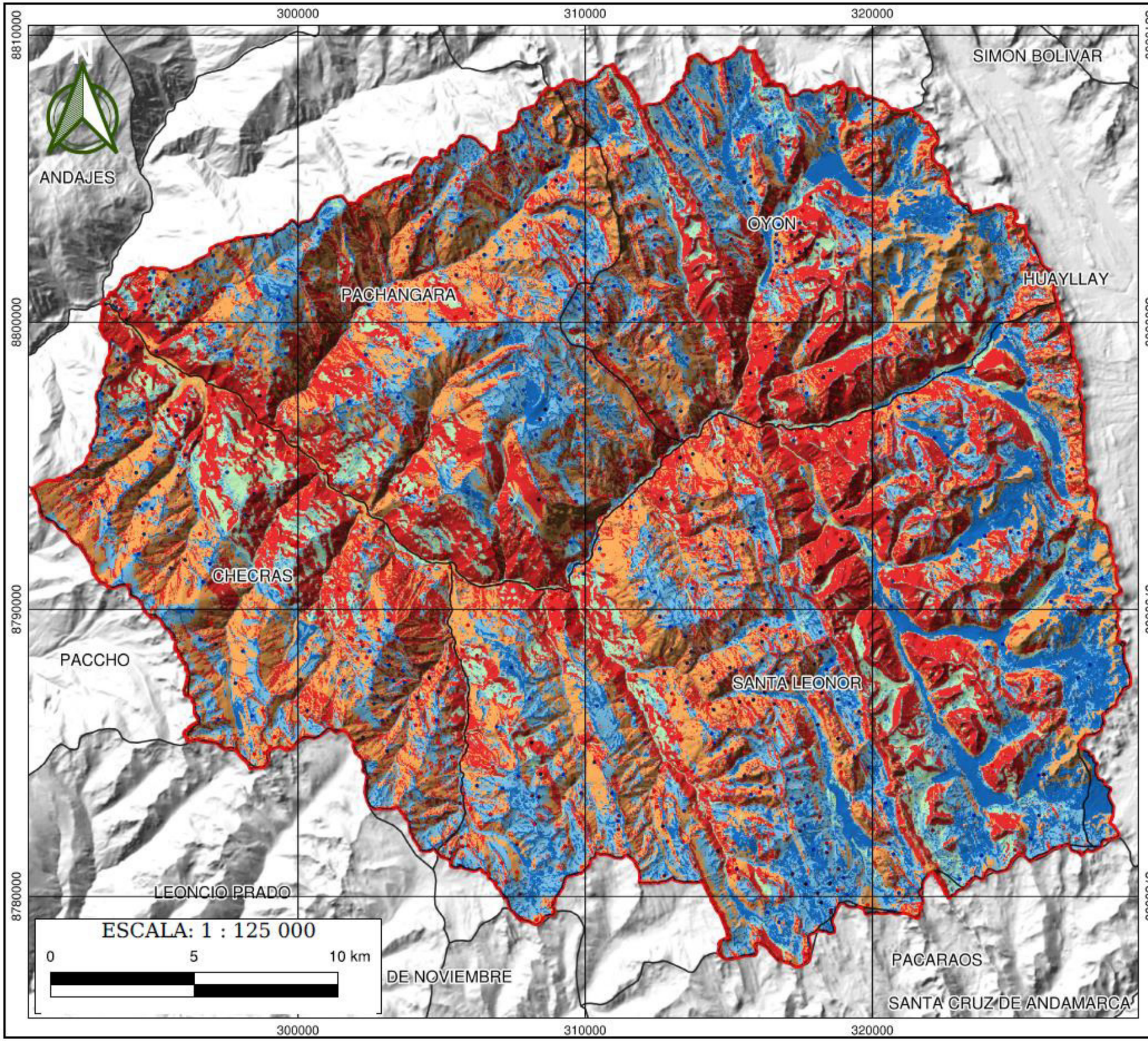
In [128…

```
# Contribución de variables [Importancia de las variables]
```

In [73]:

```
# Importancia de las variables según SVM
IV_SVM_m2 = cls_svm_m2.coef_[0]
print (IV_SVM_m2)

# Importancia de las variables según el modelo RF
IV_RF_m2 = cls_rf_m2.feature_importances_
print (IV_RF_m2)
```

```
[ 0.50142219  0.7664916  -0.09964121 -0.02931952  0.55689331  4.33475886
  6.75774155 -0.68936966]
[0.0570853  0.01760332 0.02070145 0.01888776 0.07568355 0.32424939
 0.43743545 0.04835378]
```



LEYENDA

SUB CUENCA CHECRAS

Distritos

MOVIMIENTOS EN MASA

- Movimientos en Masa [MM]
- Sin Movimientos en Masa [Sin MM]

SUSCEPTIBILIDAD A MOVIMIENTOS EN MASA

- MUY BAJA
- BAJA
- MEDIA
- ALTA
- MUY ALTA

UNIVERSIDAD NACIONAL FEDERICO VILLARREAL

FACULTAD DE INGENIERÍA GEOGRÁFICA, AMBIENTAL Y ECOTURISMO

ESCUELA PROFESIONAL DE INGENIERÍA GEOGRÁFICA

TESIS:
ESTIMACIÓN ESPACIAL DE LA SUSCEPTIBILIDAD DE MOVIMIENTOS EN MASA MEDIANTE APRENDIZAJE AUTOMÁTICO EN LA SUB CUENCA CHECRAS

SUSCEPTIBILIDAD A MOVIMIENTOS EN MASA MÁQUINA DE SOPORTE VECTORIAL PRIMER MÉTODO

AUTOR: HANSEN WIBELSMAN BUENO GÓMEZ

ASESOR: MARCO ANTONIO HERRERA DÍAZ

FUENTES: INSTITUTO GEOGRÁFICO NACIONAL (IGN), AGENCIA JAPONESA DE EXPLORACIÓN AEROESPACIAL (JAXA)

PROYECCIÓN: UTM **DATUM:** WGS84 **ZONA:** 18S

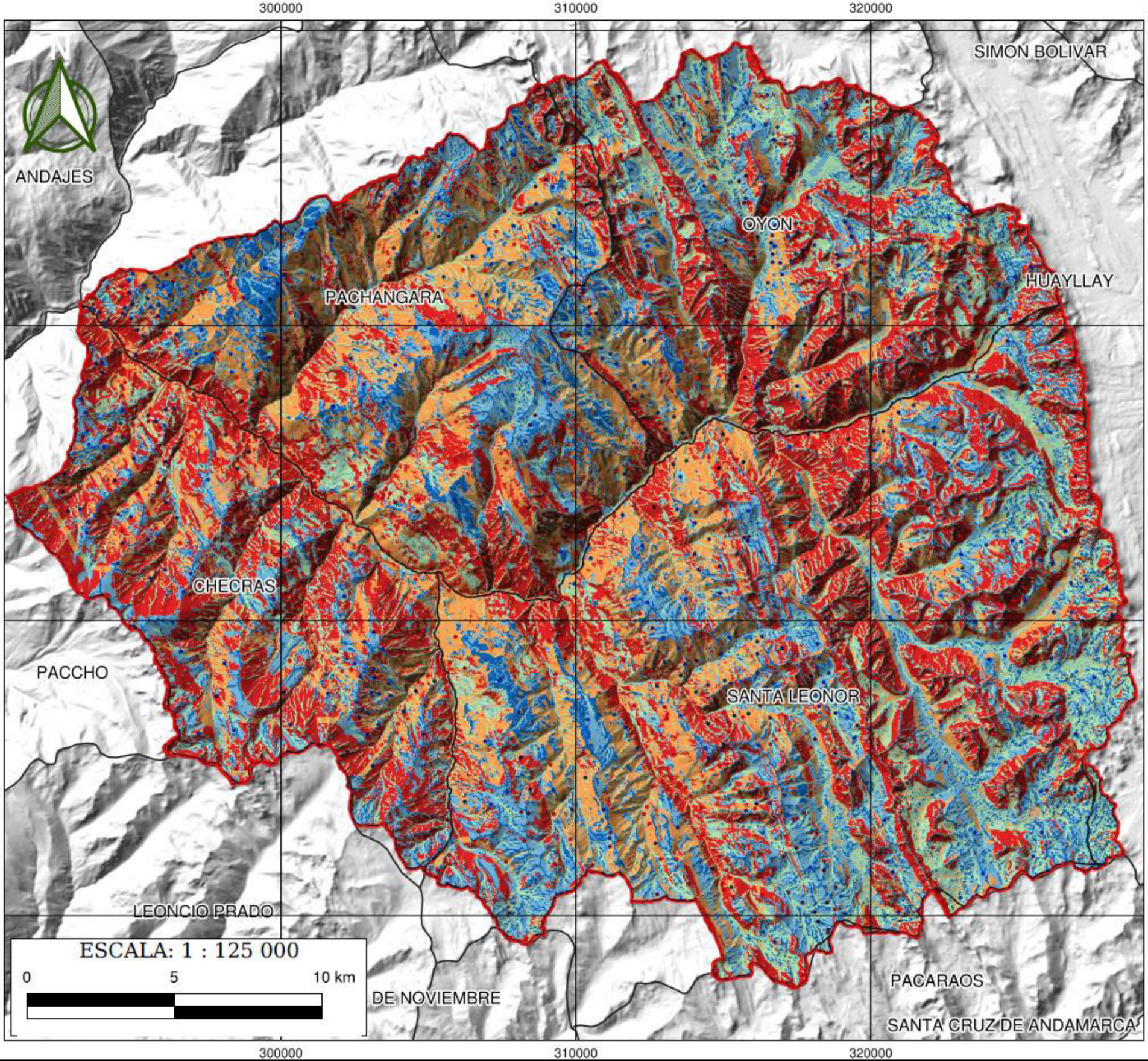
DEPTO: LIMA

PROV: HUAURA - OYON

DIST: CHECRAS - PACHANGARA - OYON - SANTA LEONOR

AÑO: 2023

MAPA **N°1**



LEYENDA

- SUB CUENCA CHECRAS
- Distritos

MOVIMIENTOS EN MASA

- Movimientos en Masa [MM]
- Sin Movimientos en Masa [Sin MM]

SUSCEPTIBILIDAD A MOVIMIENTOS EN MASA

- MUY BAJA
- BAJA
- MEDIA
- ALTA
- MUY ALTA

UNIVERSIDAD NACIONAL FEDERICO VILLARREAL

FACULTAD DE INGENIERÍA GEOGRÁFICA, AMBIENTAL Y ECOTURISMO

ESCUELA PROFESIONAL DE INGENIERÍA GEOGRÁFICA

TESIS: ESTIMACIÓN ESPACIAL DE LA SUSCEPTIBILIDAD DE MOVIMIENTOS EN MASA MEDIANTE APRENDIZAJE AUTOMÁTICO EN LA SUB CUENCA CHECRAS

SUSCEPTIBILIDAD A MOVIMIENTOS EN MASA BOSQUES ALEATORIOS PRIMER MÉTODO

AUTOR: HANSEN WIBELSMAN BUENO GÓMEZ

ASESOR: MARCO ANTONIO HERRERA DÍAZ

FUENTES: INSTITUTO GEOGRÁFICO NACIONAL (IGN), AGENCIA JAPONESA DE EXPLORACIÓN AEROESPACIAL (JAXA)

PROYECCIÓN: UTM **DATUM:** WGS84 **ZONA:** 18S

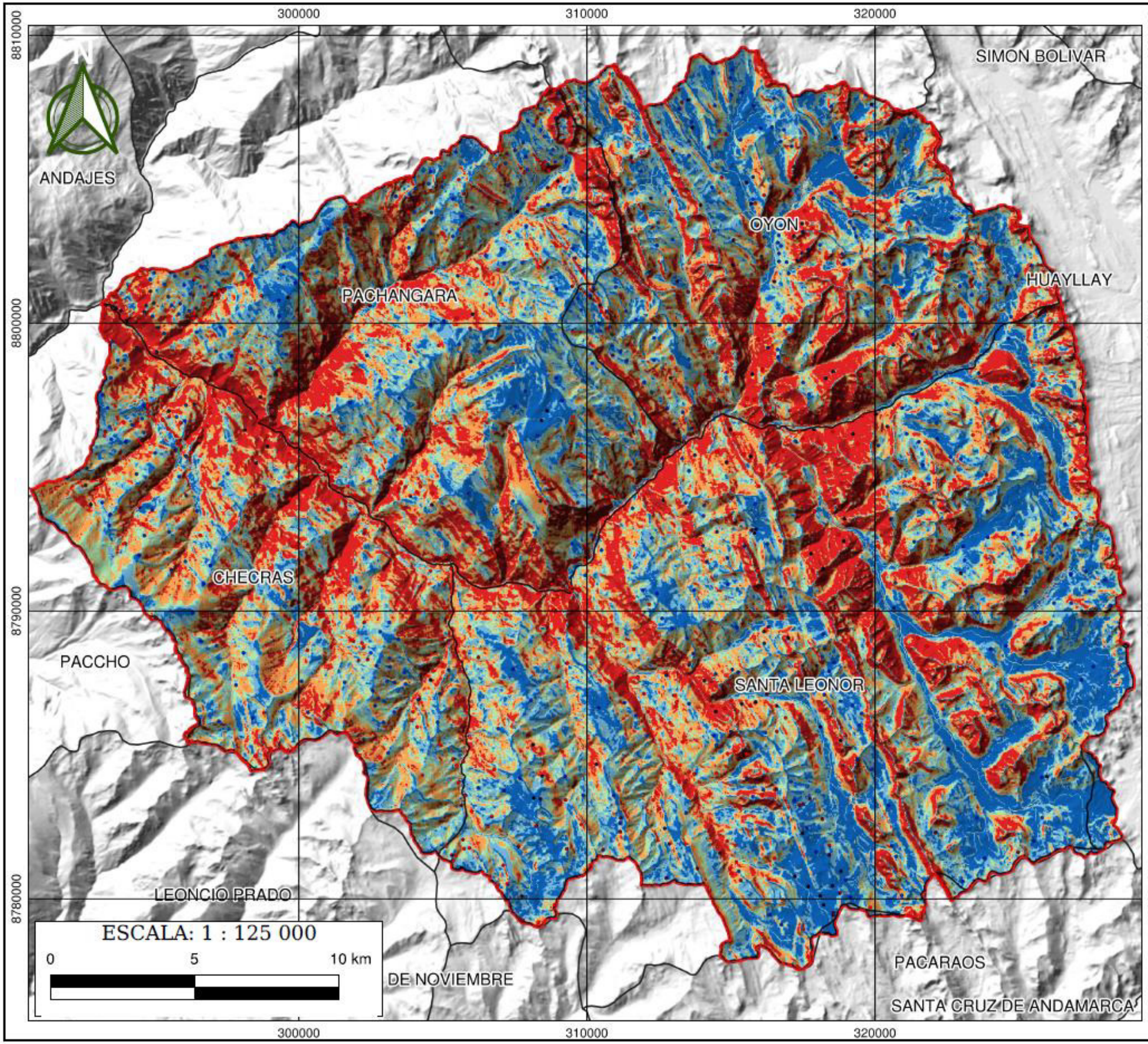
DEPTO: LIMA

PROV: HUAURA - OYON

DIST: CHECRAS - PACHANGARA - OYON - SANTA LEONOR

AÑO: 2023

MAPA N°2



LEYENDA

SUB CUENCA CHECRAS

Distritos

MOVIMIENTOS EN MASA

- Movimientos en Masa [MM]
- Sin Movimientos en Masa [Sin MM]

SUSCEPTIBILIDAD A MOVIMIENTOS EN MASA

- MUY BAJA
- BAJA
- MEDIA
- ALTA
- MUY ALTA

UNIVERSIDAD NACIONAL FEDERICO VILLARREAL

FACULTAD DE INGENIERÍA GEOGRÁFICA, AMBIENTAL Y ECOTURISMO

ESCUELA PROFESIONAL DE INGENIERÍA GEOGRÁFICA

TESIS: ESTIMACIÓN ESPACIAL DE LA SUSCEPTIBILIDAD DE MOVIMIENTOS EN MASA MEDIANTE APRENDIZAJE AUTOMÁTICO EN LA SUB CUENCA CHECRAS

SUSCEPTIBILIDAD A MOVIMIENTOS EN MASA MÁQUINA DE SOPORTE VECTORIAL SEGUNDO MÉTODO

AUTOR: HANSEN WIBELSMAN BUENO GÓMEZ

ASESOR: MARCO ANTONIO HERRERA DÍAZ

FUENTES: INSTITUTO GEOGRÁFICO NACIONAL (IGN), AGENCIA JAPONESA DE EXPLORACIÓN AEROESPACIAL (JAXA)

PROYECCIÓN: UTM **DATUM:** WGS84 **ZONA:** 18S

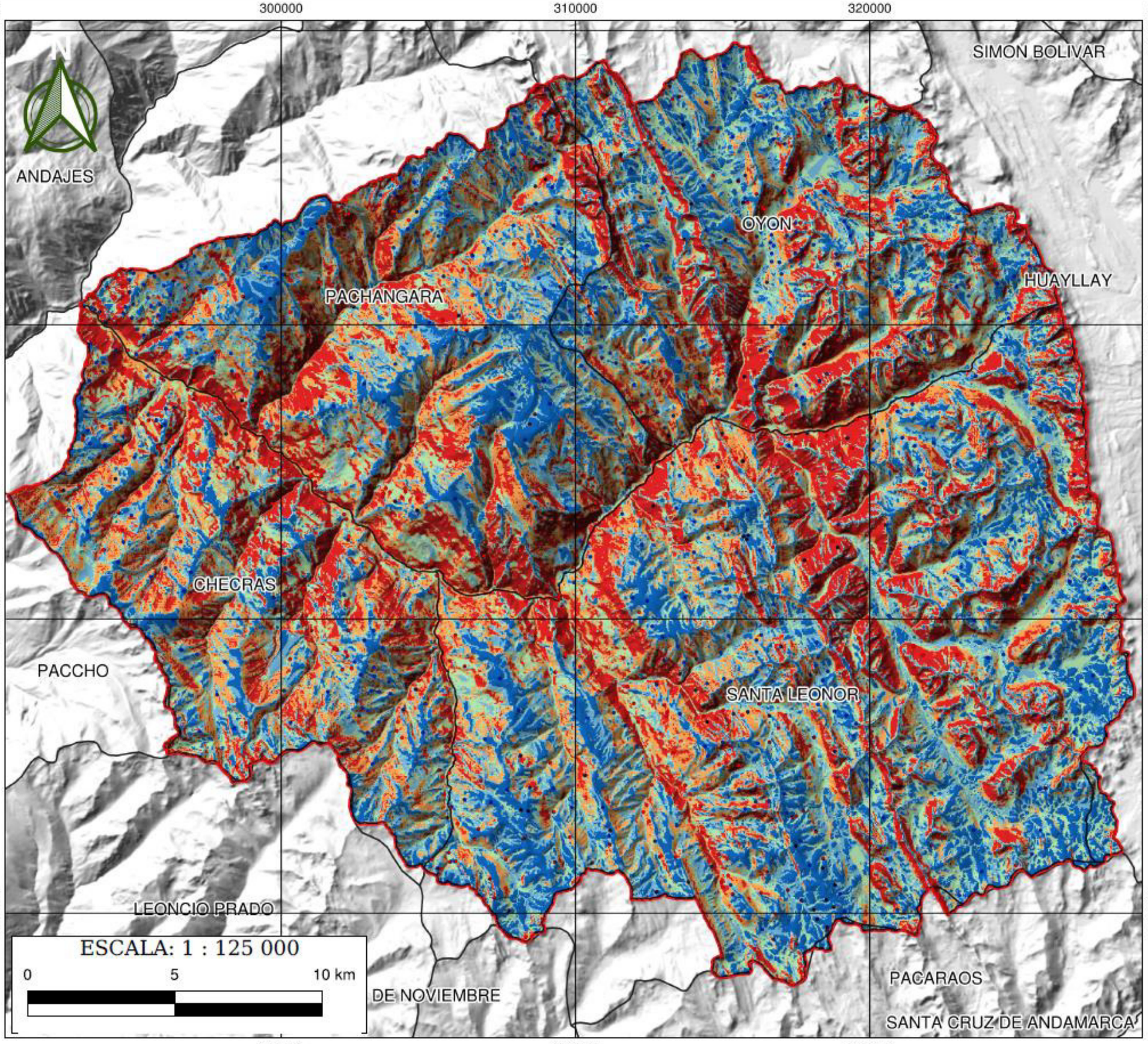
DEPTO: LIMA

PROV: HUAURA - OYON

DIST: CHECRAS - PACHANGARA - OYON - SANTA LEONOR

AÑO: 2023

MAPA N°3



LEYENDA

- SUB CUENCA CHECRAS
- Distritos

MOVIMIENTOS EN MASA

- Movimientos en Masa [MM]
- Sin Movimientos en Masa [Sin MM]

SUSCEPTIBILIDAD A MOVIMIENTOS EN MASA

- MUY BAJA
- BAJA
- MEDIA
- ALTA
- MUY ALTA

UNIVERSIDAD NACIONAL FEDERICO VILLARREAL

FACULTAD DE INGENIERÍA GEOGRÁFICA, AMBIENTAL Y ECOTURISMO

ESCUELA PROFESIONAL DE INGENIERÍA GEOGRÁFICA

TESIS: ESTIMACIÓN ESPACIAL DE LA SUSCEPTIBILIDAD DE MOVIMIENTOS EN MASA MEDIANTE APRENDIZAJE AUTOMÁTICO EN LA SUB CUENCA CHECRAS

SUSCEPTIBILIDAD A MOVIMIENTOS EN MASA BOSQUES ALEATORIOS SEGUNDO MÉTODO

AUTOR: HANSEN WIBELSMAN BUENO GÓMEZ

ASESOR: MARCO ANTONIO HERRERA DÍAZ

FUENTES: INSTITUTO GEOGRÁFICO NACIONAL (IGN), AGENCIA JAPONESA DE EXPLORACIÓN AEROESPACIAL (JAXA)

PROYECCIÓN: UTM **DATUM:** WGS84 **ZONA:** 18S

DEPTO: LIMA

PROV: HUAURA - OYON

DIST: CHECRAS - PACHANGARA - OYON - SANTA LEONOR

AÑO: 2023

MAPA N°4