Modelo Analítico Pago De Cartera

Jorge Eliecer Rojas Gómez

Trabajo de grado presentado para optar al título de

Especialista en Analítica y Ciencia de Datos

Asesora

Daniela Serna Buitrago

Magíster (MSc)

Universidad de Antioquia

Facultad de Ingeniería

Especialización en Analítica y Ciencia de Datos

Medellín, Antioquia, Colombia

2022

Cita	Rojas Gómez [1]
Referencia Estilo IEEE (2020)	[1] J. E. Rojas Gómez, "Modelo Analítico Pago De Cartera", Trabajo de grado especialización, Especialización en Analítica y Ciencia de Datos, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2022.

Especialización en Analítica y Ciencia de Datos, Cohorte III.

TABLA DE CONTENIDO

1. RESUMEN EJECUTIVO 6
2. DESCRIPCIÓN DEL PROBLEMA 7
2.1 PROBLEMA DE NEGOCIO 7
2.2 APROXIMACIÓN DESDE LA ANALÍTICA DE DATOS 7
2.3 ORIGEN DE LOS DATOS 8
2.4 MÉTRICAS DE DESEMPEÑO 8
Accuracy: 8
Matriz de confusión: 9
3. DATOS 9
3.1 DATOS ORIGINALES 9
3.2 DATASETS 12
3.3 DESCRIPTIVA 13
4. PROCESO DE ANALÍTICA 18
4.1 PIPELINE PRINCIPAL 18
4.2 PREPROCESAMIENTO 19
4.2.1 Exploración inicial 19
4.2.2 Modificación inicial de variables 19
4.2.3 Limpieza de datos 20
4.2.4 Correlación y datos faltantes 20
4.2.5 Codificación final de variables 21
4.3 MODELOS 22
LogisticRegression: Regresión Logística 22
DecisionTreeClassifier: Árboles de decisión para clasificación 24
KNeighborsClassifier 25

KMeans 26
5. METODOLOGÍA 28
5.1 VALIDACIÓN 29
5.3 ITERACIONES y EVOLUCIÓN 30
5.4 HERRAMIENTAS 31
6. RESULTADOS 32
6.1 MÉTRICAS 32
6.1.1 Matriz de confusión: 32
6.1.2 Accuracy y Balanced Accuracy: 33
7. CONCLUSIONES 35
8. BIBLIOGRAFIA 36

LISTA DE TABLAS

Tabla 1 Listado de variables datos originales 12
Tabla 2 Muestra de datos originales 13
Tabla 3 Descripción de variables numéricas 13
Tabla 4 Matriz correlación de variables 20
Tabla 5 Descripción de variables set de datos final 22
Tabla 6 Tabla resultado entrenamiento modelo regresión 23
Tabla 7 Tabla resultado evaluación modelo regresión 23
Tabla 8 Tabla métricas modelo regresión 24
Tabla 9 Tabla resultado entrenamiento modelo árboles 24
Tabla 10 Tabla métricas modelo árboles 25
Tabla 11 Tabla métricas modelo KNeighbors 26
Tabla 12 Tabla resultados entrenamiento modelo Kmeans 27
Tabla 13 Tabla métricas modelo Kmeans 27
Tabla 14 DataSet para procesamiento de modelos 29

Tabla 15 Muestra del VectorAssembler para el set de datos 29
Tabla 16 Resultados de matrices de confusión para los modelos en estudio. 33
Tabla 17 Valores resultantes accuracy y balanced_accuracy para los modelos en estudio 33

LISTA DE FIGURAS

Figura 1 Descripción Matriz de confusión 9
Figura 2 Descripción de variables datos originales 12
Figura 3 Gráfico de tendencia y valores atípicos datos originales 14
Figura 4 Visualización información variable de interés “Línea de Crédito” 14
Figura 5 Gráfico variable fuente colocación 15
Figura 6 Gráfico variable mes 15
Figura 7 Gráfico variable Municipio Posconflicto 15
Figura 8 Gráfico variable género 16
Figura 9 Gráfico variable Líneas de Producción 16
Figura 10 Gráfico comparativo Linea de Credito y Fuente Colocacion 17
Figura 11 Gráfico relación Líneas de Crédito y de Producción 17
Figura 12 Gráfico comparativo Linea de Credito y Municipio PosConflicto 17
Figura 13 Gráfico comparativo Línea de Crédito y Género 18
Figura 14 Gráfico relación variables Línea de Crédito y Mes 18
Figura 15 Flujo proceso analítico 19
Figura 16 Mapa de calor para la matriz de correlación 21
Figura 17 Ciclo Metodología Desarrollo 28
Figura 18 Gráfico comparativo resultado de métricas para modelos en estudio 34

1. RESUMEN EJECUTIVO

Las entidades financieras privadas, microfinancieras, los fondos públicos, las cooperativas y aseguradoras de servicios brindan constantemente facilidades de otorgamiento de créditos y servicios a una población variada con la finalidad de atraer más clientes. Estas entidades dependen en gran parte del recaudo recibido en contraprestación de los créditos con lo cual incrementa sus utilidades y con ello aumentar las inversiones en diferentes sectores.

En un caso particular de los fondos públicos, existe FINAGRO como la entidad financiera de desarrollo para el sector agropecuario y rural colombiano que otorga recursos a través de los intermediarios financieros (bancos, cooperativas e intermediarios microfinancieras) para que estos a su vez asignen créditos a los empresarios del campo en apoyo al desarrollo de proyectos productivos. Pero surge la pregunta, ¿qué tanto de estos recursos públicos otorgados por el gobierno por medio de FINAGRO cumplen el ciclo completo?, entendiendo al ciclo como el proceso que contempla las etapas de estudio, asignación, desembolso y retorno (pago de cuotas); con lo cual se asegurar que la inversión tenga un retorno positivo para las entidades y por ende para el gobierno.

Para lograr mitigar el riesgo de generar créditos que de alguna forma u otra no retornan la inversión y permitan aumentar el recurso disponible, las entidades financieras han implementado modelos clasificatorios predictivos que ayuden en este objetivo. Tomando los datos publicados por FINAGRO en el año 2021 y con la realización de una fase inicial de exploración, se encuentra un dataset con buena calidad de información para trabajar.

Ya en una fase de ejecución de los modelos de clasificación, se buscaron las métricas necesarias y suficientes para corroborar cuál de los modelos seleccionados para el estudio, lograba cumplir con el comportamiento esperado. Dichas métricas cambiaron durante las diferentes interacciones, dado que era necesario que se pudieran aplicar de una forma equitativa para los modelos que se encontraban en comparación.

Finalmente dados los resultados obtenidos por medio de las métricas de accuracy, balanced accuracy y matriz de confusión, se toman los modelos de DecisionTreeClassifier y KNeighborsClassifier, como los de mejor desempeño para el estudio académico que se efectuó.

El repositorio con el código de los notebooks y la fuente de datos que se usaron para el desarrollo del proyecto se encuentra en el siguiente enlace: https://github.com/jorkrojas/MonografiaEACD .

2. DESCRIPCIÓN DEL PROBLEMA

En la actualidad las entidades con fondos públicos, entidades financieras privadas y aseguradoras de servicios brindan facilidades de otorgamiento de créditos y servicios a una población variada con la finalidad de atraer más clientes e incrementar sus utilidades. Sin embargo, el retorno o recuperación de cartera se vuelve una tarea complicada y tediosa para los negocios.

Es por este motivo que cada vez más entidades buscan la implementación de modelos predictivos que le permitan disminuir el riesgo de conceder créditos y/o servicios a clientes que acorde a sus características no paguen lo adeudado.

2.1 PROBLEMA DE NEGOCIO

El fondo para el financiamiento del Sector Agropecuario (FINAGRO), es la entidad financiera de desarrollo para el sector agropecuario y rural colombiano que otorga recursos a través de los intermediarios financieros (bancos, cooperativas e intermediarios microfinancieras) para que estos a su vez otorguen créditos a los empresarios del campo en apoyo al desarrollo de su proyecto productivo. El presente proyecto con base a los créditos otorgados por FINAGRO en el año 2021 y mediante el análisis clasificatorio predictivo, tiene como objetivo lograr establecer si el cliente pagará o no el crédito o servicio adquirido con la entidad. Este análisis se realizará probando diferentes modelos de Machine Learning para la clasificación y verificar cuál de estos se adapta mejor al comportamiento de la fuente de información. [1, 2]

2.2 APROXIMACIÓN DESDE LA ANALÍTICA DE DATOS

Se elaborará un proceso analítico basado en mínimo tres tipos diferentes de modelos de Machine Learning para clasificación de datos. Inicialmente se realizará un análisis exploratorio de la información, en cuanto a sus variables y cantidad de registros.

Una vez se tengan los datos limpios y con el análisis inicial, se va dar inicio a probar los diferentes algoritmos de clasificación contenidos en las librerías de Python para Machine Learning. Estos algoritmos nos deberán dar una serie de resultados que se analizarán para determinar cuál de ellos otorga mejor comportamiento

2.3 ORIGEN DE LOS DATOS

Se obtienen los datos de la página https://www.datos.gov.co/, donde se encuentran datos abiertos lo cual permite descubrir, acceder y utilizar información pública bajo licencia abierta y sin restricciones legales para su aprovechamiento. Esto reglamentado en Colombia, con la Ley 1712 de 2014 de la Ley de Transparencia y del Derecho de Acceso a la Información Pública Nacional.

Para el caso se toma el recurso referente a los "Colocaciones de Crédito Sector Agropecuario – 2021", https://www.datos.gov.co/Agricultura-y-Desarrollo-Rural/Colocaciones-de-Cr-dito-Sector-Agropecuario-2021/w3uf-w9ey. [1]

2.4 MÉTRICAS DE DESEMPEÑO

Las métricas de desempeño a utilizar en el modelo se encuentran sujetas a los algoritmos de clasificación de Machine Learning, que sean seleccionados y si la misma métrica permite una evaluación equitativa entre los modelos. Algunas métricas que se estuvieron evaluando fueron:

- Índice de la silueta
- Accuracy (Precisión)
- Balanced Accuracy
- Área bajo la curva de funcionamiento del receptor (ROC) (AUC)
- Matriz de confusión o error

Acorde a los resultados preliminares, se toma la decisión de implementar Accuray y Matriz de confusión como métricas de desempeño, dado que ofrecen para la evaluación entre los modelos una mejor equidad. [2, 3]

Accuracy:

Se define como una métrica para evaluar los modelos de clasificación. Haciendo referencia al porcentaje total de elementos clasificados correctamente. Formalmente se calcula de la siguiente manera:

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

Matriz de confusión:

Es una tabla que describe el rendimiento de un modelo de Machine Learning en los datos. Su nombre se da porque hace fácil detectar dónde el sistema está confundiendo dos clases.

True Positives (TP): cuando la clase real del punto de datos era 1 (Verdadero) y la predicha es también 1 (Verdadero)

Verdaderos Negativos (TN): cuando la clase real del punto de datos fue 0 (Falso) y el pronosticado también es 0 (Falso).

False Positives (FP): cuando la clase real del punto de datos era 0 (False) y el pronosticado es 1 (True).

False Negatives (FN): Cuando la clase real del punto de datos era 1 (Verdadero) y el valor predicho es 0 (Falso).

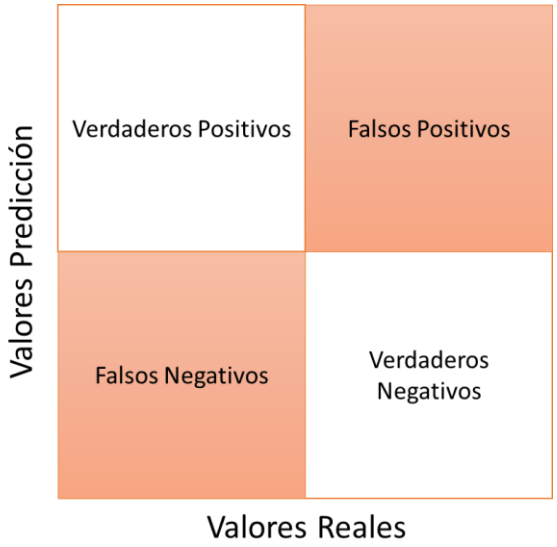


Figura 1 Descripción Matriz de confusión

3. DATOS

3.1 DATOS ORIGINALES

En este apartado se describen los datos originales que se tienen para realizar el proceso analítico. Como bien se mencionó anteriormente, se tendrá el recurso “Colocaciones de Crédito Sector Agropecuario – 2021”, correspondiente con las operaciones de crédito colocadas en el sector agropecuario 2021. El recurso se encuentra en un formato .CSV, pero si es necesario está disponible en la página de datos.gov.co en los formatos de RDF, RSS, XML y TSV.

En la tabla siguiente se muestra el listado de variables (columnas) de las que se dispone en la fuente de información. De igual forma para cada variable se tiene una descripción y el tipo de dato que almacena.

Nombre de Columna	Descripción	Tipo
Año	Corresponde al año en el que se generó la operación	Número
Mes	Corresponde al número del mes del año (1= Enero, 2=Febrero,….12=Diciembre) en el que se generó la operación	Número
fuente Colocacion	Fuente de Fondeo de las operaciones de credito (Redescuento, Agropecuaria, Sustituta)	Texto simple
Id Tipo Prod	Codigo correpondiente a la clasificación de productores de FINAGRO	Número
Tipo Productor	Nombre correspondiente a la clasificación de productores de FINAGRO	Texto simple
Valor Inversion	Valor total del proyecto a cargo del productor.	Número
Colocacion	Valor del desembolso de la operación. (Valor del crédito)	Número
ID Depto	Codigo del Departamento de Colombia segun DANE, en donde se ejecutó el proyecto	Número
Departamento Inversion	Nombre del Departamento de Colombia según DANE, donde se ejecutó el proyecto.	Texto simple
Id Munic	Codigo del Municipio de Colombia segun DANE, en donde se ejecutó el proyecto	Número
Municipio Inversion	Nombre del Municipio de Colombia según DANE, donde se ejecutó el proyecto.	Texto simple
Municipio de PostConflico?	Hace referencia, si el Municipio de Inversión donde se ejecutó el proyecto pertenece al Conjunto de Municipios afectados por la violencia Clasificados como municipios Postcoinflicto	Texto simple

Nombre de Columna	Descripción	Tipo
DEPCOL	Codigo del Departamento de Colombia segun DANE, en donde se otorgó el credito destinado al proyecto	Número
Departamento de Colocacion de Credito	Nombre del Departamento de Colombia según DANE, en donde se otorgó el credito destinado al proyecto	Texto simple
MUNCOL	Codigo del Municipio de Colombia segun DANE, en donde se otorgó el credito destinado al proyecto	Número
Municipio Colocacion de Credito	Nombre del Municipio de Colombia según DANE, en donde se otorgó el credito destinado al proyecto	Texto simple
Plazo	Número de meses para el pago de la operación.	Número
Linea de Credito	Clasificación de las actividades financiables (Capital de Trabajo, Inversión, Normalización de Cartera)	Texto simple
Linea de Produccion	Clasificación de las actividades financiables para cada Linea de Credito	Texto simple
ID Rubro	Corresponde al codigo del destino asignado para la actividad financiada y registrada en FINAGRO.	Número
Destino de Credito	Corresponde al nombre del destino asignado para la actividad financiada y registrada en FINAGRO.	Texto simple
Genero	Sigla del género de la persona, "S= Persona Juridica", "H=Hombre", "M=Mujer"	Texto simple
% FAG	Porcentaje de la Garantía FAG que respalda la operación desembolsada, No todas las operaciones tienen garantía	Número
Vlr Inic Garantia	Valor inicialmente garantízado por el FAG (Valor de laColocación por el % de Garantía FAG)	Número
LATITUD	Coordenadas de georeferciación del centro del Municipio de Inversión	Número
LONGITUD	Coordenadas de georeferciación del centro del Municipio de Inversión	Número

Nombre de Columna	Descripción	Tipo
CANTIDAD	Valor de control.	Número

Tabla 1 Listado de variables datos originales [1]

3.2 DATASETS

Acorde a la fuente de información, se realiza el cargue del archivo al notebook, donde se realizó exploración de los datos. [3, 4]

Por medio de la librería de pandas, el archivo .CVS es convertido en un dataset con las características:

- 27 columnas
- 4799978 registros

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 479978 entries, 0 to 479977
Data columns (total 27 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   Año                                   479978 non-null  object
 1   Mes                                   479978 non-null  int64
 2   fuente Colocacion                     479978 non-null  object
 3   Id Tipo Prod                          479978 non-null  int64
 4   Tipo Productor                        479978 non-null  object
 5   Valor Inversion                       479978 non-null  object
 6   Colocacion                            479978 non-null  object
 7   ID Depto                              479978 non-null  int64
 8   Departamento Inversion                479978 non-null  object
 9   Id Munic                              479978 non-null  object
 10  Municipio Inversion                   479978 non-null  object
 11  Municipio de PostConflico?            479978 non-null  object
 12  DEPCOL                                479978 non-null  int64
 13  Departamento de Colocacion de Credito 479978 non-null  object
 14  MUNCOL                                479978 non-null  object
 15  Municipio Colocacion de Credito       479978 non-null  object
 16  Plazo                                 479978 non-null  int64
 17  Linea de Credito                      479978 non-null  object
 18  Linea de Produccion                   479978 non-null  object
 19  ID Rubro                              479978 non-null  object
 20  Destino de Credito                    479978 non-null  object
 21  Genero                                479978 non-null  object
 22  % FAG                                 301147 non-null  float64
 23  Vlr Inic Garantia                     301147 non-null  object
 24  LATITUD                               479978 non-null  float64
 25  LONGITUD                              479978 non-null  float64
 26  CANTIDAD                              479978 non-null  int64
dtypes: float64(3), int64(6), object(18)
```

Figura 2 Descripción de variables datos originales

Realizando un muestreo de los datos, se puede apreciar la información contenida en cada uno de las variables.

	Año	Mes	fuente Colocacion	Id Tipo Prod	Tipo Productor	Valor Inversion	Colocacion	ID Depto	Departamento Inversion	Id Munic	...	Linea de Credito	Linea de Produccion	ID Rubro	Destino de Credito	Genero	% FAG	Vlr Inic Garantia	LATITUD	LONGITUD	CANTIDAD
0	2,021	2	REDESCUENTO	1	MEDIANO	25000000.0	11185899.0	15	BOYACÁ	15599	...	Normalización de Cartera	SIEMBRAS (I)	141550.0	RAMIRIQUÍ	M	NaN	NaN	5.416667	-73.333333	1
1	2,021	2	REDESCUENTO	1	MEDIANO	53600000.0	40000000.0	19	CAUCA	19548	...	Inversión	INFRAEST Y ADECU DE TIERRAS (I)	347050.0	PIENDAMÓ - TUNÍA	H	NaN	NaN	2.750000	-76.500000	1
2	2,021	2	REDESCUENTO	1	MEDIANO	107759000.0	100000000.0	5	ANTIOQUIA	5237	...	Inversión	INFRAEST Y ADECU DE TIERRAS (I)	347495.0	DONMATÍAS	H	NaN	NaN	6.500000	-75.333333	1
3	2,021	2	AGROPECUARIA	0	PEQUEÑO	2950000.0	2089560.0	76	VALLE DEL CAUCA	76233	...	Normalización de Cartera	SOSTENIMIENTO (CT)	160000.0	CALI	M	NaN	NaN	3.660278	-76.692778	1
4	2,021	2	AGROPECUARIA	0	PEQUEÑO	7700000.0	6801690.0	13	BOLÍVAR	13670	...	Normalización de Cartera	MAQUINARIA Y EQUIPO (I)	447350.0	SAN PABLO	H	NaN	NaN	10.052778	-75.268056	1

5 rows × 27 columns

Tabla 2 Muestra de datos originales

3.3 DESCRIPTIVA

Con la información a analizar contenida en un dataset, se realiza una descripción general de la información. Es de esta forma que utilizando la función describe() de pandas, se muestran cálculos estadísticos para las variables numéricas.

	Mes	Id Tipo Prod	ID Depto	DEPCOL	Plazo	% FAG	LATITUD	LONGITUD	CANTIDAD
count	479978.000000	479978.000000	479978.000000	479978.00000	479978.000000	301147.000000	479978.000000	479978.000000	479978.0
mean	6.697232	0.167533	38.750620	38.15367	44.873746	80.716188	5.089111	-74.905629	1.0
std	3.380452	0.434091	25.422284	25.46692	31.929417	10.210081	2.475267	1.557761	0.0
min	1.000000	0.000000	5.000000	5.00000	1.000000	1.000000	-4.215278	-81.750000	1.0
25%	4.000000	0.000000	17.000000	15.00000	18.000000	80.000000	3.166667	-76.000000	1.0
50%	7.000000	0.000000	25.000000	25.00000	36.000000	80.000000	5.200000	-75.083333	1.0
75%	10.000000	0.000000	66.000000	63.00000	60.000000	80.000000	6.464167	-73.612778	1.0
max	12.000000	2.000000	99.000000	99.00000	240.000000	100.000000	12.576855	-67.046459	1.0

Tabla 3 Descripción de variables numéricas

De igual manera, para ver la tendencia y valores atípicos, se elabora un gráfico que caja (boxplot).

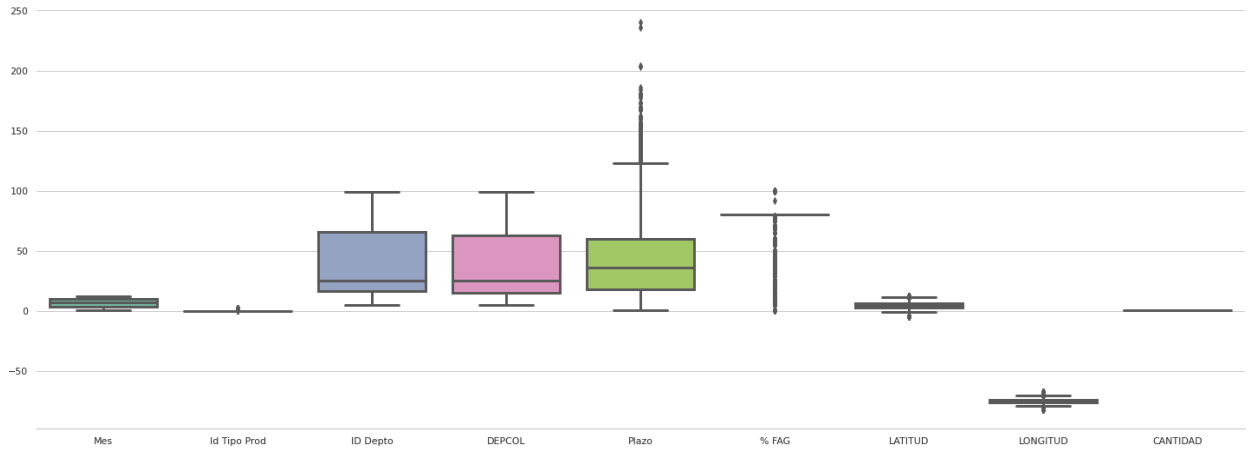
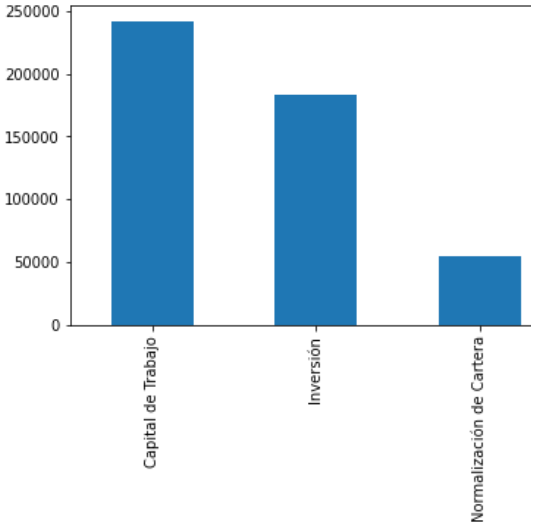


Figura 3 Gráfico de tendencia y valores atípicos datos originales

Para el estudio analico clasificatorio que se realiza, se toma como variable de interés las "Líneas de Crédito", a continuación se aprecian los valores generales para la variable:



Linea de Credito	
Capital de Trabajo	242165
Inversión	183125
Normalización de Cartera	54688

Figura 4 Visualización información variable de interés "Línea de Crédito"

De igual forma se hace un muestreo general de otras variables de forma gráfica, como el mes, la fuente de colocación del crédito, el género de la persona que toma el crédito, si el municipio donde se emite el crédito es de postconflicto y la línea de producción de la persona con lo que soporta el crédito.

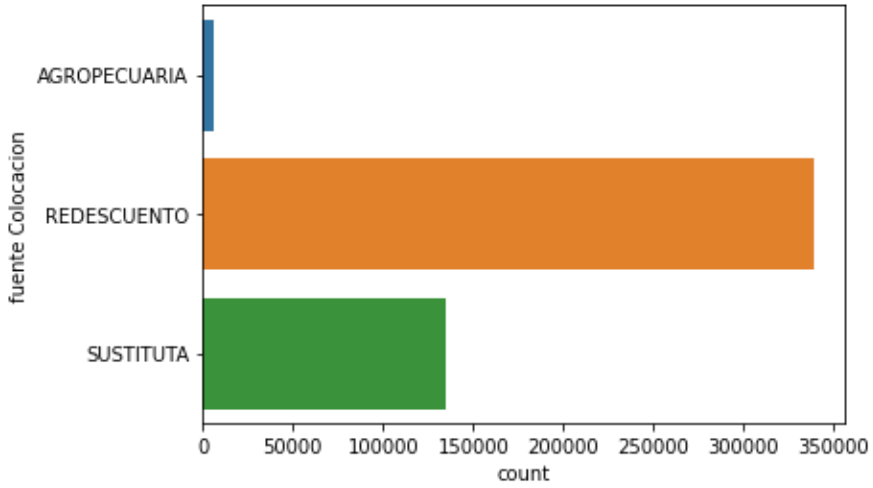


Figura 5 Gráfico variable fuente colocación

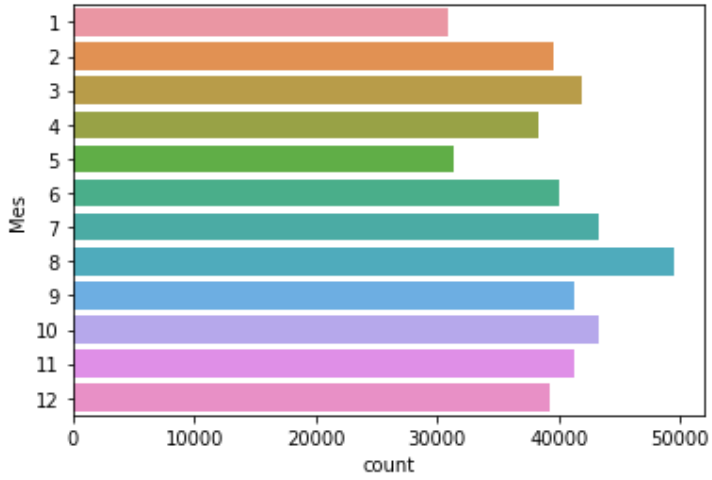


Figura 6 Gráfico variable mes

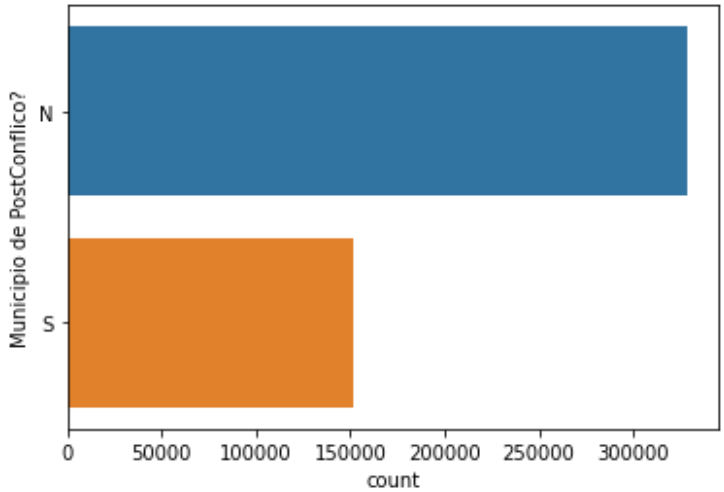


Figura 7 Gráfico variable Municipio Posconflicto

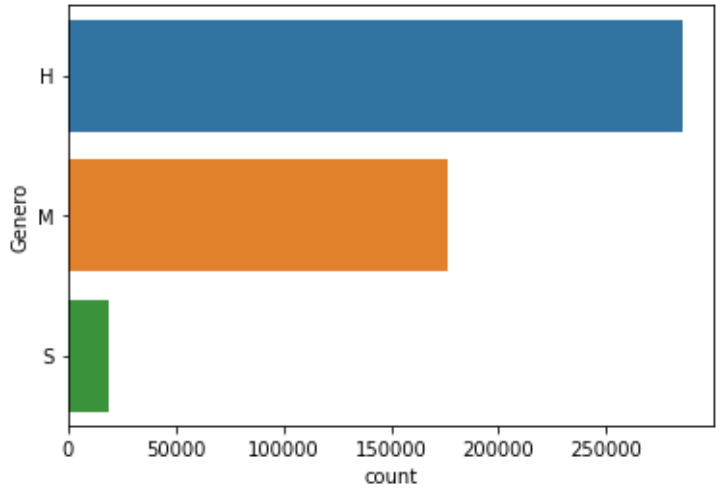


Figura 8 Gráfico variable género

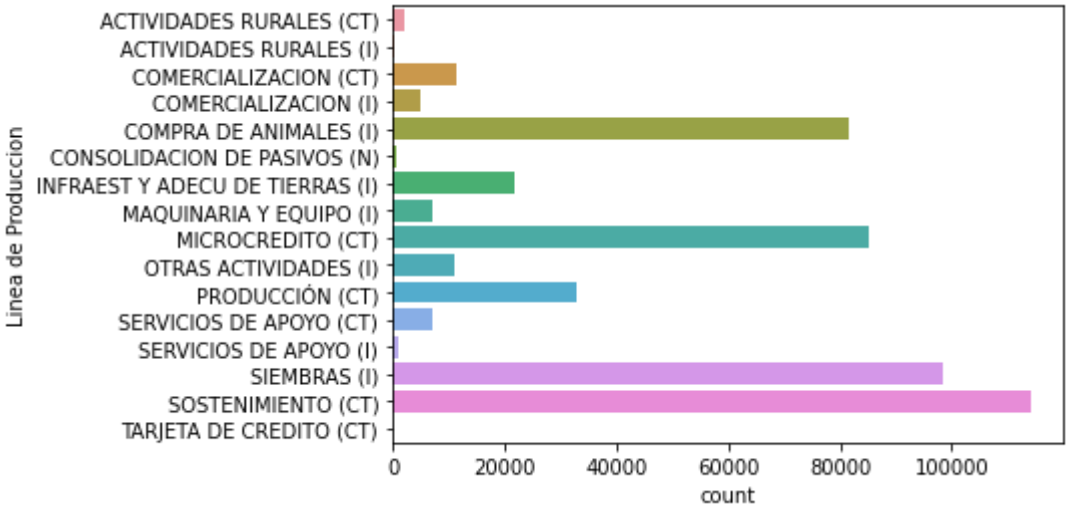


Figura 9 Gráfico variable Líneas de Producción

Finalmente, para tener una visión general de la variable de interés, se elaboran gráficos comparativos de las “Líneas de Crédito” con otras variables.

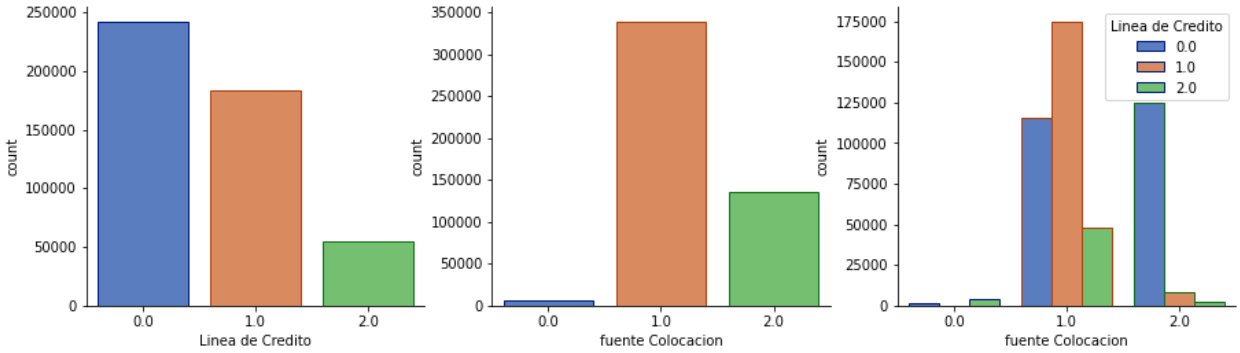


Figura 10 Gráfico comparativo Linea de Credito y Fuente Colocacion

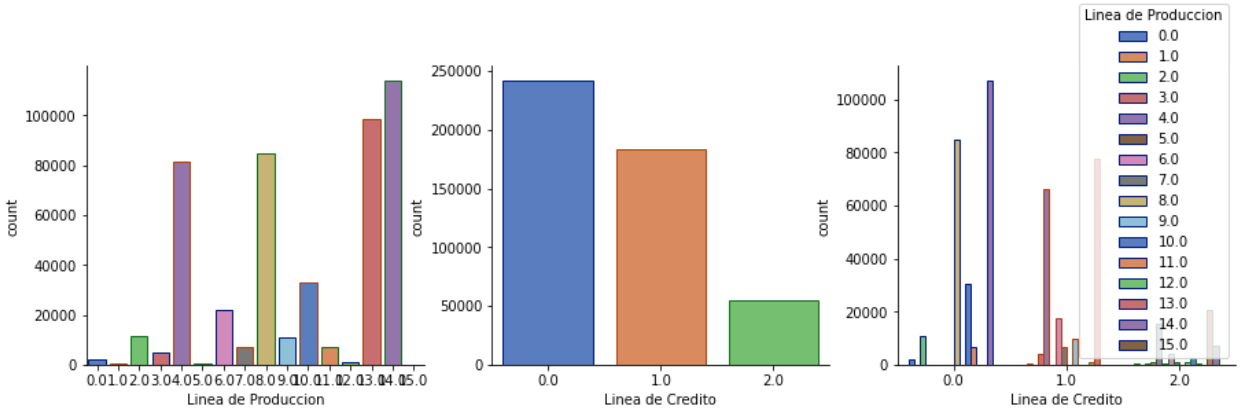


Figura 11 Gráfico relación Líneas de Crédito y de Producción

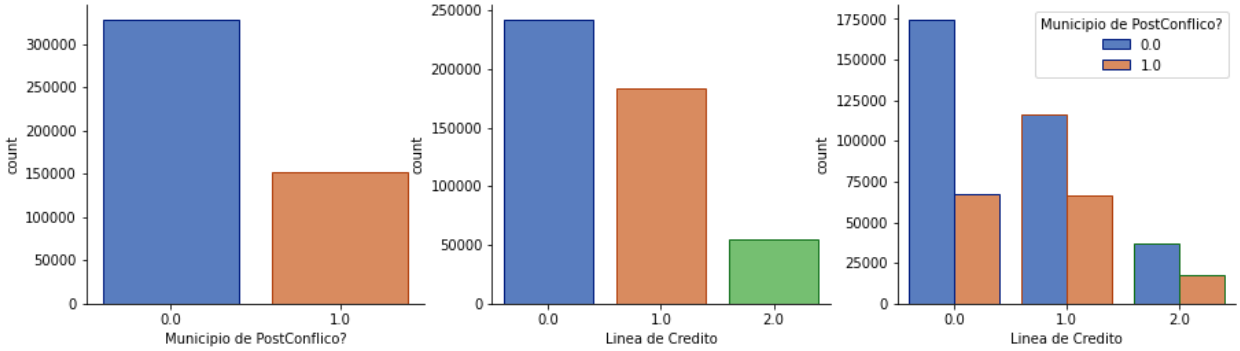


Figura 12 Gráfico comparativo Linea de Credito y Municipio PosConflicto

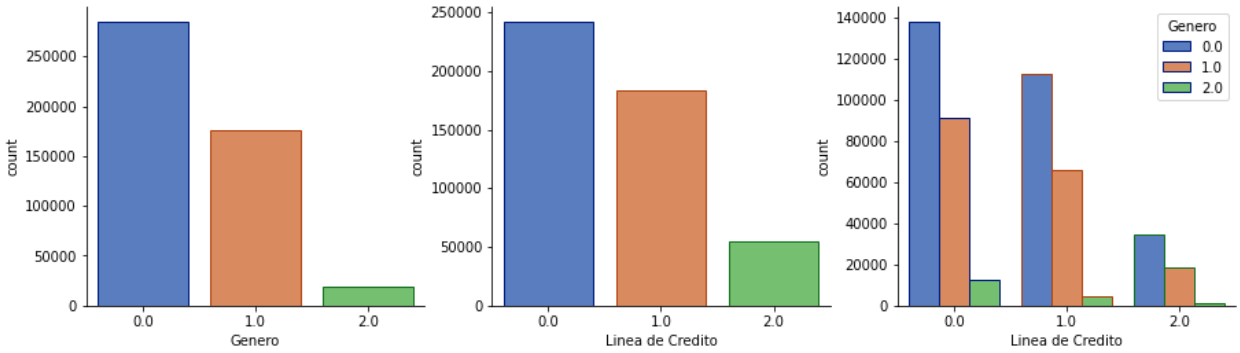


Figura 13 Gráfico comparativo Línea de Crédito y Género

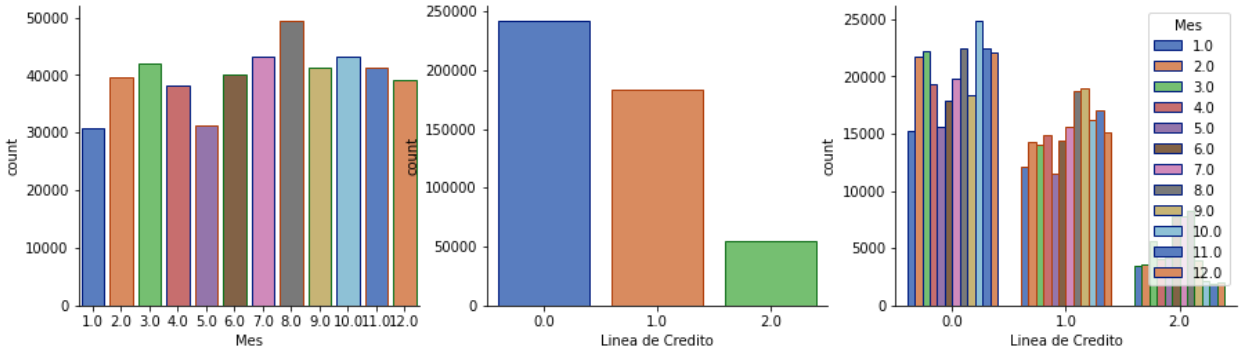


Figura 14 Gráfico relación variables Línea de Crédito y Mes

4. PROCESO DE ANALÍTICA

4.1 PIPELINE PRINCIPAL

En el siguiente diagrama se plantea el flujo del proceso para el tratamiento y analitica aplicada al conjunto de datos perteneciente al problema del negocio descrito.

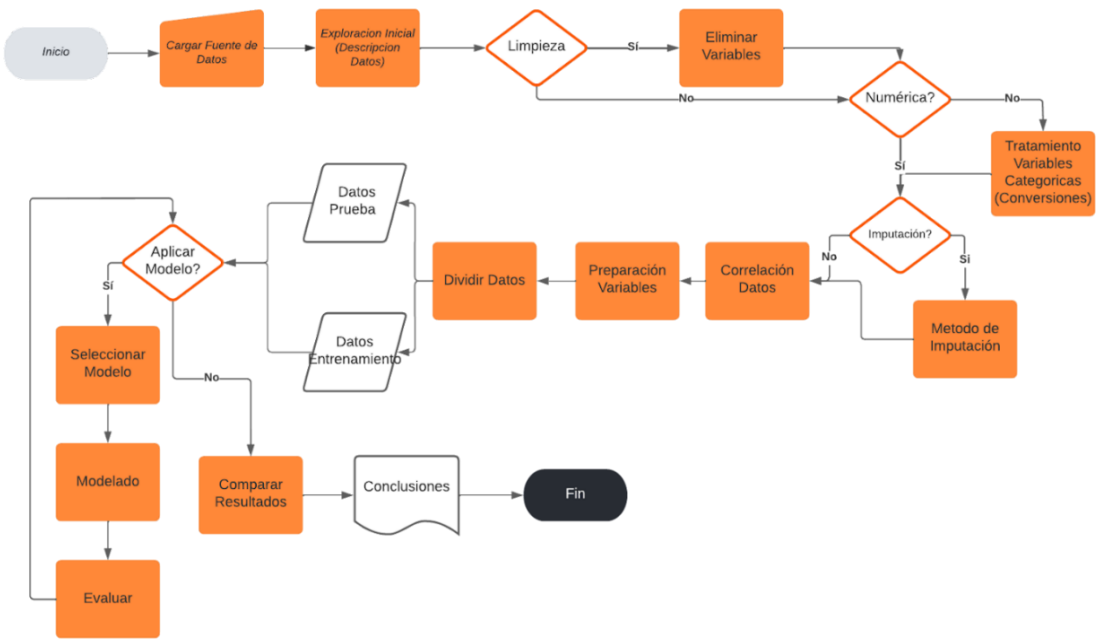


Figura 15 Flujo proceso analítico

4.2 PREPROCESAMIENTO

Como se hizo mención anteriormente, el dataset fue tomado de datos libres, perteneciente a los créditos otorgados por FINAGRO en el año 2021. La descripción de campos es de forma general y no se tiene la posibilidad de contactar con una persona experta que pueda aportar en el estudio. Es por este motivo que para el preprocesamiento y con la finalidad de conocer de forma clara el set de datos, se ejecutaron una serie de etapas, las cuales se describen a continuación:

4.2.1 Exploración inicial

Tomando el set de datos original, se aplica el análisis exploratorio, donde la descripción e identificación de variables permite conocer los tipos de datos, los formatos en los cuales se encuentran, la longitud de cada columna y las constantes que se manejan en el set de datos. Con esta exploración de la información se preseleccionan las variables de interés para la elaboración del estudio clasificatorio.

4.2.2 Modificación inicial de variables

Posterior a la exploración inicial, es necesario realizar la transformación de las variables, para ello se ejecutan los procesos de:

- Conversión de variables tipo object a categóricas.
- Cambio de cadenas de texto a numéricas.

4.2.3 Limpieza de datos

Durante esta etapa se realiza la eliminación de variables mencionadas a continuación, que no son de interés y no aportan en el proceso.

- La variable Año corresponde a un valor constante único 2021 para todos los registros.
- La variable Cantidad es 1 en todos los registros dado que es un valor de control.
- Los Identificadores de tipo producto, id departamento, id municipio y rubro, se consideran suficientes al proceso y por ende se eliminan las variables categóricas que contienen la misma información, como los nombres de los departamentos y municipios.
- La ubicación geográfica, no son variables de interés para este estudio.

4.2.4 Correlación y datos faltantes

Con el set de datos manipulado con la eliminación de variables, se procede a ejecutar la correlación de las variables actuales.

	Mes	Id Tipo Prod	Valor Inversion	Colocacion	ID Depto	Id Munic	Plazo	ID Rubro	% FAG
Mes	1.000000	-0.013361	-0.003142	-0.002439	0.004354	0.004418	0.006567	0.019652	0.011232
Id Tipo Prod	-0.013361	1.000000	0.145307	0.156017	0.026556	0.025183	-0.148058	0.178657	-0.356087
Valor Inversion	-0.003142	0.145307	1.000000	0.812689	-0.006820	-0.007114	-0.004417	0.046433	-0.038706
Colocacion	-0.002439	0.156017	0.812689	1.000000	-0.010020	-0.010356	-0.008008	0.047463	-0.117346
ID Depto	0.004354	0.026556	-0.006820	-0.010020	1.000000	0.999943	-0.009323	-0.032593	-0.069189
Id Munic	0.004418	0.025183	-0.007114	-0.010356	0.999943	1.000000	-0.008668	-0.032769	-0.068869
Plazo	0.006567	-0.148058	-0.004417	-0.008008	-0.009323	-0.008668	1.000000	0.194841	0.272947
ID Rubro	0.019652	0.178657	0.046433	0.047463	-0.032593	-0.032769	0.194841	1.000000	0.032888
% FAG	0.011232	-0.356087	-0.038706	-0.117346	-0.069189	-0.068869	0.272947	0.032888	1.000000

Tabla 4 Matriz correlación de variables

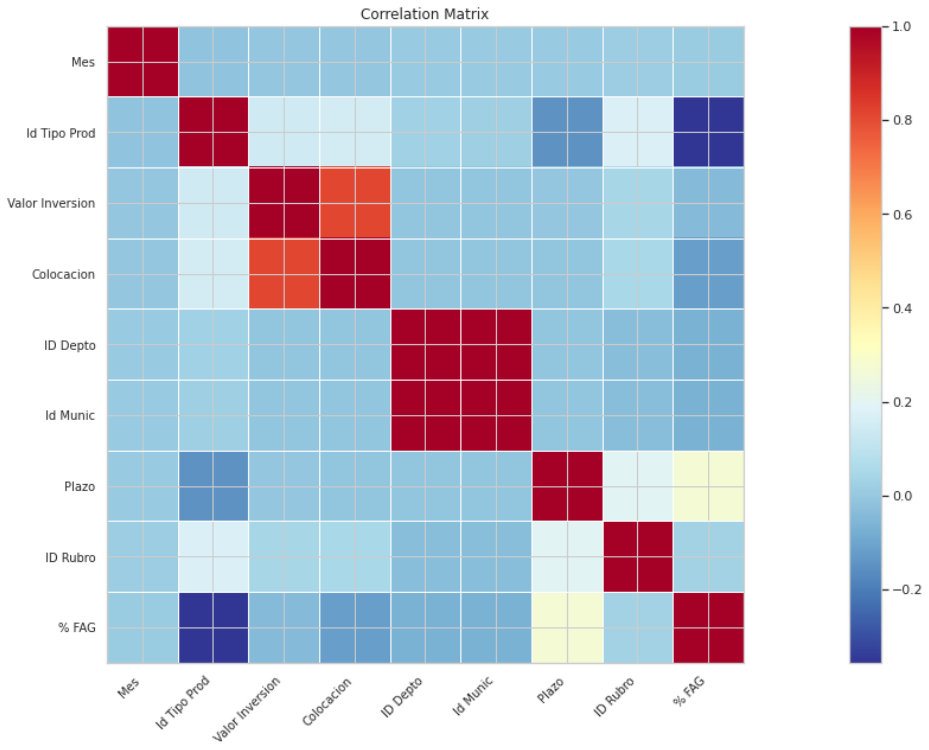


Figura 16 Mapa de calor para la matriz de correlación

Finalmente para terminar la etapa, con el apoyo de la librería *from sklearn.impute import MissingIndicator*, son identificadas las variables donde hay datos faltantes y calcular el porcentaje de los mismos.

```
print("Características donde hay datos faltantes: ", Indicador.features_)
```

```
Características donde hay datos faltantes:  [13 14]
```

```
POS = np.where(Datos_Indicador == True) # Se busca aquellos datos que tiene un valor Booleanos igual a True
print("Porcentaje de Datos Faltantes (%): ", 100*(len(POS[0])/(Datos_Indicador.shape[0]*Datos_Indicador.shape[1])))
```

```
Porcentaje de Datos Faltantes (%):  37.258165999274965
```

Dado el resultado correspondiente con un 37% de datos faltantes para las variables “% FAG” y “Vlr Inic Garantia”. Se toma la determinación de prescindir de dichas variables, dado que su aporte al estudio no es significativo.

4.2.5 Codificación final de variables

Finalmente con apoyo de las librerías *from sklearn.preprocessing import LabelEncoder* para la codificación de etiquetas y *from collections import defaultdict* para generar un diccionario

nuevo de valores, se realiza la conversión de variables categóricas a valores numéricos de forma aleatoria, excepto de la variable objetivo para la cual se realiza una conversión definida.

```
#Se codifican las categorias de la variable objetivo
df_DataSetNum["Linea de Credito"]=df_DataSetNum["Linea de Credito"].replace({"Capital de Trabajo": 0, "Inversión": 1, "Normalización de Cartera":2})
```

Realizado este proceso se tiene el set de datos con la información requerida para el tratamiento con los modelos de clasificación.

```
root
 |-- Mes: double (nullable = true)
 |-- fuente Colocacion: double (nullable = true)
 |-- Id Tipo Prod: double (nullable = true)
 |-- Valor Inversion: double (nullable = true)
 |-- Colocacion: double (nullable = true)
 |-- ID Depto: double (nullable = true)
 |-- Id Munic: double (nullable = true)
 |-- Municipio de PostConflico?: double (nullable = true)
 |-- Plazo: double (nullable = true)
 |-- Linea de Credito: double (nullable = true)
 |-- Linea de Produccion: double (nullable = true)
 |-- ID Rubro: double (nullable = true)
 |-- Genero: double (nullable = true)

+---+----------------+-----------+---------------+-----------+--------+--------+--------------------------+-----+----------------+-------------------+--------+------+
|Mes|fuente Colocacion|Id Tipo Prod|Valor Inversion| Colocacion|ID Depto|Id Munic|Municipio de PostConflico?|Plazo|Linea de Credito|Linea de Produccion|ID Rubro|Genero|
+---+----------------+-----------+---------------+-----------+--------+--------+--------------------------+-----+----------------+-------------------+--------+------+
|2.0|             1.0|        1.0|          2.5E7|1.1185899E7|    15.0| 15599.0|                       0.0| 36.0|             2.0|               13.0|141550.0|   1.0|
|2.0|             1.0|        1.0|         5.36E7|      4.0E7|    19.0| 19548.0|                       0.0| 84.0|             1.0|                6.0|347050.0|   0.0|
|2.0|             1.0|        1.0|      1.07759E8|      1.0E8|     5.0|  5237.0|                       1.0| 60.0|             1.0|                6.0|347495.0|   0.0|
|2.0|             0.0|        0.0|      2950000.0|  2089560.0|    76.0| 76233.0|                       1.0|  9.0|             2.0|               14.0|160000.0|   1.0|
|2.0|             0.0|        0.0|      7700000.0|  6801690.0|    13.0| 13670.0|                       1.0| 46.0|             2.0|                7.0|447350.0|   0.0|
+---+----------------+-----------+---------------+-----------+--------+--------+--------------------------+-----+----------------+-------------------+--------+------+
only showing top 5 rows
```

Tabla 5 Descripción de variables set de datos final

4.3 MODELOS

Con la finalidad de cumplir con los objetivos planteados en el modelo analitico pago de cartera, se realiza la consulta y adaptación de diferentes modelos de clasificación, los cuales serán descritos a continuación:

LogisticRegression: Regresión Logística

En la regresión logística se calcula la ecuación de la recta que mejor represente al conjunto de datos con el fin de predecir, pero teniendo en cuenta que la variable a predecir es categórica y puede tener n categorías, lo que propone es calcular las ecuaciones de n rectas, una para cada categoría. Enseguida se describe el código utilizado para el entrenamiento y evaluación del modelo. [2, 3]

Entrenamiento Modelo:

```
from pyspark.ml.classification import LogisticRegression
lr = LogisticRegression(labelCol="Linea de Credito", featuresCol="features",maxIter=10)
```

```
lr_model=lr.fit(trainingData)
trainingSummary = lr_model.summary
trainingSummary.predictions.select('Linea de Credito','prediction','probability').show(5)
```

```
+----------------+----------+--------------------+
|Linea de Credito|prediction|         probability|
+----------------+----------+--------------------+
|             2.0|       2.0|[0.28735745514429...|
|             2.0|       0.0|[0.60102193467237...|
|             2.0|       0.0|[0.64194528128714...|
|             2.0|       0.0|[0.48717951896455...|
|             2.0|       2.0|[0.17658982005679...|
+----------------+----------+--------------------+
only showing top 5 rows
```

Tabla 6 Tabla resultado entrenamiento modelo regresión

Evaluación Modelo:

```
predict_testRL=lr_model.transform(testData)
predict_testRL.select('Linea de Credito','prediction','probability').show(10)
```

```
+----------------+----------+--------------------+
|Linea de Credito|prediction|         probability|
+----------------+----------+--------------------+
|             2.0|       0.0|[0.50129849310800...|
|             2.0|       2.0|[0.28776899892488...|
|             2.0|       1.0|[0.04565082018276...|
|             2.0|       0.0|[0.64764946018309...|
|             2.0|       1.0|[0.04708671692367...|
|             2.0|       1.0|[0.07409477692230...|
|             2.0|       1.0|[0.01203779239303...|
|             2.0|       2.0|[0.34395108707989...|
|             2.0|       1.0|[0.05365785326851...|
|             2.0|       1.0|[0.07655659877891...|
+----------------+----------+--------------------+
only showing top 10 rows
```

Tabla 7 Tabla resultado evaluación modelo regresión

```
clase_real_RL = predict_testRL.select(['Linea de Credito']).collect()
clase_prediccion_RL = predict_testRL.select(['prediction']).collect()

print("Medidas de error")
print(classification_report(clase_real_RL, clase_prediccion_RL))
```

Medidas de error

	precision	recall	f1-score	support
0.0	0.87	0.97	0.91	72413
1.0	0.84	0.94	0.89	55010
2.0	0.39	0.03	0.06	16287
accuracy			0.85	143710
macro avg	0.70	0.65	0.62	143710
weighted avg	0.80	0.85	0.81	143710

Tabla 8 Tabla métricas modelo regresión

DecisionTreeClassifier: Árboles de decisión para clasificación

Con los árboles de decisión la variable a predecir es categórica y se basa a partir del conjunto histórico de datos con los cuales construye en árbol que tiene en sus nodos una pregunta sobre alguno de los atributos y en las hojas alguna de las categorías de la variable objetivo. Al igual que en el modelo anterior a continuación se describe el código utilizado para el entrenamiento y evaluación del modelo.

Entrenamiento Modelo:

```
from pyspark.ml.classification import DecisionTreeClassifier
dt = DecisionTreeClassifier(labelCol="Linea de Credito", featuresCol="features")
dt_model=dt.fit(trainingData)
predict_trainAD = dt_model.transform(trainingData)
predict_trainAD.select('Linea de Credito','prediction').show(5)
```

```
+----------------+----------+
|Linea de Credito|prediction|
+----------------+----------+
|             2.0|       2.0|
|             2.0|       0.0|
|             2.0|       0.0|
|             2.0|       2.0|
|             2.0|       2.0|
+----------------+----------+
only showing top 5 rows
```

Tabla 9 Tabla resultado entrenamiento modelo árboles

Evaluación del modelo:

```
predict_testAD = dt_model.transform(testData)
clase_realAD = predict_testAD.select(['Linea de Credito']).collect()
```

```
clase_prediccionAD = predict_testAD.select(['prediction']).collect()


print("Medidas de error")
print(classification_report(clase_realAD, clase_prediccionAD))
```

Medidas de error

	precision	recall	f1-score	support
0.0	0.96	1.00	0.98	72413
1.0	0.92	0.97	0.95	55010
2.0	0.86	0.55	0.67	16287
accuracy			0.94	143710
macro avg	0.91	0.84	0.87	143710
weighted avg	0.94	0.94	0.93	143710

Tabla 10 Tabla métricas modelo árboles

KNeighborsClassifier

Definido como el algoritmo K vecinas más cercanos, es considerado dentro de los algoritmos de aprendizaje supervisado que son aplicados para la solución de problemas de clasificación y regresión. A continuación, se describe el código utilizado para el entrenamiento y evaluación del modelo.

Entrenamiento Modelo:

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler


X_trainKC, X_testKC, y_trainKC, y_testKC = train_test_split(Xkc, ykc, random_state=0)
scaler = MinMaxScaler()
X_trainKC = scaler.fit_transform(X_trainKC)
X_testKC = scaler.transform(X_testKC)


from sklearn.neighbors import KNeighborsClassifier


n_neighbors = 7


knn = KNeighborsClassifier(n_neighbors)
knn.fit(X_trainKC, y_trainKC)
```

Evaluación del modelo:

```
predKNN = knn.predict(X_testKC)
print("Medidas de error")
print(classification_report(y_testKC, predKNN))
```

Medidas de error

	precision	recall	f1-score	support
0.0	0.96	0.99	0.98	60865
1.0	0.91	0.97	0.94	45548
2.0	0.83	0.53	0.65	13582
accuracy			0.93	119995
macro avg	0.90	0.83	0.85	119995
weighted avg	0.93	0.93	0.92	119995

Tabla 11 Tabla métricas modelo KNeighbors

KMeans

Este modelo divide el conjunto de datos en un número predefinido de grupos k. Es el método más comúnmente utilizado, la idea del método es definir k centroides, uno por clúster, y los datos son asociados al centroide más cercano. Siguiendo el esquema de los modelos mencionados con anterioridad, enseguida se describe el código utilizado para el entrenamiento y evaluación del modelo.

Entrenar Modelo:

```
from pyspark.ml.clustering import KMeans
km = KMeans( featuresCol='features', k=3, predictionCol='cluster', distanceMeasure='euclidean')
km_model = km.fit(trainingData)
predictionsKM = km_model.transform(trainingData)
predictionsKM.distinct().show()
```

features	Linea de Credito	cluster
[1.0,1.0,0.0,1750...	0.0	0
[1.0,1.0,0.0,1850...	0.0	0
[1.0,1.0,0.0,1900...	2.0	0
[1.0,1.0,0.0,2000...	0.0	0
[1.0,1.0,0.0,2400...	0.0	0
[1.0,1.0,0.0,2580...	0.0	0
[1.0,1.0,0.0,3000...	0.0	0
[1.0,1.0,0.0,3000...	0.0	0
[1.0,1.0,0.0,3200...	0.0	0
[1.0,1.0,0.0,3210...	0.0	0
[1.0,1.0,0.0,3460...	0.0	0
[1.0,1.0,0.0,5000...	0.0	0
[1.0,1.0,0.0,5190...	0.0	0
[1.0,1.0,0.0,6000...	2.0	0
[1.0,1.0,0.0,6000...	0.0	0
[1.0,1.0,0.0,6000...	1.0	0
[1.0,1.0,0.0,6000...	1.0	0
[1.0,1.0,0.0,6000...	0.0	0
[1.0,1.0,0.0,6000...	0.0	0
[1.0,1.0,0.0,6500...	0.0	0

only showing top 20 rows

Tabla 12 Tabla resultados entrenamiento modelo Kmeans

Evaluación Modelo:

```
from pyspark.ml.evaluation import ClusteringEvaluator
evaluator = ClusteringEvaluator(predictionCol='cluster')
clase_realKM = predictionsKM.select(['Linea de Credito']).collect()
clase_prediccionKM = predictionsKM.select(['cluster']).collect()


print("Medidas de error")
print(classification_report(clase_realKM, clase_prediccionKM))
```

Medidas de error

	precision	recall	f1-score	support
0.0	0.50	1.00	0.67	169752
1.0	0.00	0.00	0.00	128115
2.0	0.53	0.00	0.00	38401
accuracy			0.50	336268
macro avg	0.34	0.33	0.22	336268
weighted avg	0.32	0.50	0.34	336268

Tabla 13 Tabla métricas modelo Kmeans

5. METODOLOGÍA

El desarrollo del proyecto se realiza siguiendo un ciclo de mejora continua, el cual permite realizar modificaciones incorporando o eliminando segmentos al notebook. Las etapas que componen la metodología de desarrollo, incorporan el proceso analítico mencionado anteriormente.

- Lectura de Datos: Cargue inicial de la fuente de información para trabajar. Es la base de datos original.
- Depuración y Limpieza: Etapa donde se eliminan, agregan o modifican las variables que se tienen de la fuente de información.
- Análisis y Mejoras: Es el inicio y fin del ciclo de mejora continua de la metodología, es la etapa donde se toman las decisiones y se implementan cambios para mejorar el proceso.
- Preparación: Etapa donde se cargan las librerías necesarias para el modelamiento, durante esta se realiza un análisis general de las variables que fueron elegidas para el procesamiento de los modelos y se realiza la correspondiente división entre datos de entrenamiento y de prueba.
- Modelado: Consiste en la construcción de cada uno de los modelos de clasificación que se van a probar con la data.
- Evaluación: Es el resultado que se tiene del modelado, validando estos con las métricas elegidas que permitan equitativamente comparar los modelos.

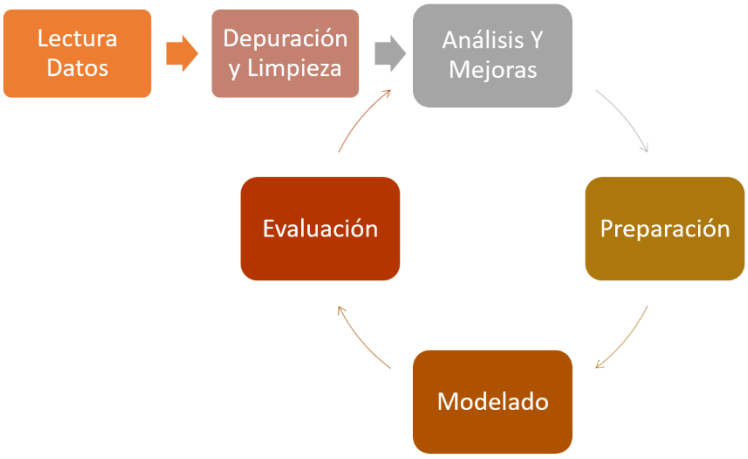


Figura 17 Ciclo Metodología Desarrollo

5.1 VALIDACIÓN

Al dataset luego de depuración y limpieza se le hace una partición, donde se deja el 70% de los datos para el entrenamiento y del 30% restante para pruebas .

```
[49] df_ModeloV2=sqlCtx.createDataFrame(df_DataSetModelo)
     df_ModeloV2.printSchema()
     df_ModeloV2.show(5)

root
 |-- Mes: double (nullable = true)
 |-- fuente Colocacion: double (nullable = true)
 |-- Id Tipo Prod: double (nullable = true)
 |-- Valor Inversion: double (nullable = true)
 |-- Colocacion: double (nullable = true)
 |-- ID Depto: double (nullable = true)
 |-- Id Munic: double (nullable = true)
 |-- Municipio de PostConflico?: double (nullable = true)
 |-- Plazo: double (nullable = true)
 |-- Linea de Credito: double (nullable = true)
 |-- Linea de Produccion: double (nullable = true)
 |-- ID Rubro: double (nullable = true)
 |-- Genero: double (nullable = true)

+---+----------------+-----------+--------------+----------+--------+--------+--------------------------+-----+---------------+------------------+--------+------+
|Mes|fuente Colocacion|Id Tipo Prod|Valor Inversion| Colocacion|ID Depto|Id Munic|Municipio de PostConflico?|Plazo|Linea de Credito|Linea de Produccion|ID Rubro|Genero|
+---+----------------+-----------+--------------+----------+--------+--------+--------------------------+-----+---------------+------------------+--------+------+
|2.0|             1.0|        1.0|         2.5E7|1.1185899E7|    15.0| 15599.0|                       0.0| 36.0|            2.0|              13.0|141550.0|   1.0|
|2.0|             1.0|        1.0|        5.36E7|      4.0E7|    19.0| 19548.0|                       0.0| 84.0|            1.0|               6.0|347050.0|   0.0|
|2.0|             1.0|        1.0|     1.07759E8|      1.0E8|     5.0|  5237.0|                       1.0| 60.0|            1.0|               6.0|347495.0|   0.0|
|2.0|             0.0|        0.0|     2950000.0|  2089560.0|    76.0| 76233.0|                       1.0|  9.0|            2.0|              14.0|160000.0|   1.0|
|2.0|             0.0|        0.0|     7700000.0|  6801690.0|    13.0| 13670.0|                       1.0| 46.0|            2.0|               7.0|447350.0|   0.0|
+---+----------------+-----------+--------------+----------+--------+--------+--------------------------+-----+---------------+------------------+--------+------+
only showing top 5 rows
```

Tabla 14 DataSet para procesamiento de modelos

```
vectorAssembler = VectorAssembler(inputCols = \
    ['Mes', 'fuente Colocacion', 'Id Tipo Prod', 'Valor Inversion', 'Colocacion', 'ID Depto', 'Id Munic', 'Municipio de PostConflico?', 'Plazo', 'Linea de Produccion', 'ID Rubro', 'Genero'], outputCol = 'features')
vest = vectorAssembler.transform(df_ModeloV2)
vest = vest.select(['features', 'Linea de Credito'])
vest.show(5)
```

```
+--------------------+----------------+
|            features|Linea de Credito|
+--------------------+----------------+
|[2.0,1.0,1.0,2.5E...|             2.0|
|[2.0,1.0,1.0,5.36...|             1.0|
|[2.0,1.0,1.0,1.07...|             1.0|
|[2.0,0.0,0.0,2950...|             2.0|
|[2.0,0.0,0.0,7700...|             2.0|
+--------------------+----------------+
only showing top 5 rows
```

Tabla 15 Muestra del VectorAssembler para el set de datos

(trainingData, testData) = vest.randomSplit([0.7, 0.3])

trainingData = 336268

testData = 143710

5.3 ITERACIONES y EVOLUCIÓN

Durante el desarrollo del proyecto se ejecutaron una serie de interacciones, algunas llevaron un mayor tiempo y otras se hicieron de forma exploratoria. Durante esta primera interacción no se probaron los modelos, dado que, al no tener una persona con el conocimiento del negocio, se toma la decisión de tomar un tiempo prudente para analizar la fuente de información.

A continuación, se hace mención únicamente de las interacciones que tuvieron mayor influencia en el proceso del proyecto:

- Iteración 0:
  Exploración inicial de la fuente de datos por medio de análisis descriptivo.
- Iteración 1:
  Depuración y limpieza de datos. Basados en la exploración realizada se toman las decisiones sobre las variables que se eliminarán y las que se manejan como primordiales.
- Iteración 2:
  Visualización de información, para tener claridad de los datos con los cuales se contará, se realizan una serie de gráficos comparativos.
- Iteración 3:
  Elección de los modelos de clasificación que van a ser utilizados. Regresión logística, árboles de decisión y máquinas de soporte.
- Iteración 4:
  Selección de de métricas de evaluación. Accuracy, área bajo la curva, Matriz de confusión, índice de silueta.
- Iteración 4:
  Preparación y división del dataset para entrenamiento y pruebas.
- Iteración 5:
  Codificación de modelo regresión logística

- Iteración 6:
  Codificación modelo árboles de decisión para clasificación.
- Iteración 7:
  Codificación modelo de máquinas de soporte. Se presentan dificultades para la implementación del modelo en mención. Dadas las dificultades se opta por eliminar el modelo y elegir otros modelos.
- Iteración 8:
  Codificación modelos KMeans y KNeighbors.
- Iteración 9:
  Implementación de segmentos de evaluación para cada uno de los modelos.
- Iteración 10:
  Análisis de resultados con las métricas. No se logra implementar la métrica área bajo la curva e índice de silueta para todos los modelos, por tanto, se toma la decisión de comparar los resultados de los modelos con Accuracy y matriz de confusión.
- Iteración 11:
  Visualización y comparación de resultados para las métricas de evaluación.
- Iteración 12:
  Implementación y comparación de resultados de la métrica balanced accuracy.
- Iteración 13:
  Conclusiones al procesamiento y modelado.

5.4 HERRAMIENTAS

El proyecto se realiza usando las herramientas:

1. Python: Como lenguaje de programación y procesamiento de la información, por medio de las librerías asociadas.
2. Jupyter Notebook: Herramienta para el procesamiento del notebook de forma local.
3. Google Colab: Herramienta para realizar las pruebas online del notebook.
4. Google drive: Repositorio para el almacenamiento del notebook, documentos y fuente de información.
5. Github: Repositorio elegido para almacenar el notebook final desarrollado.

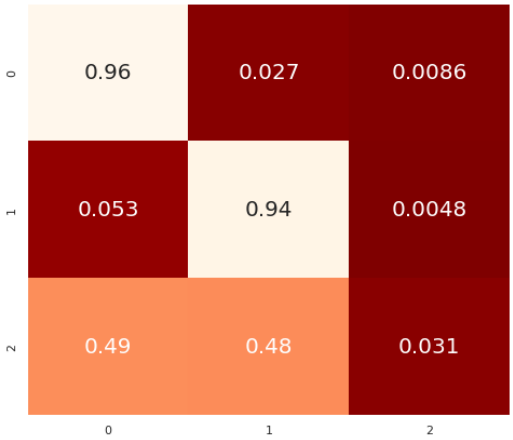
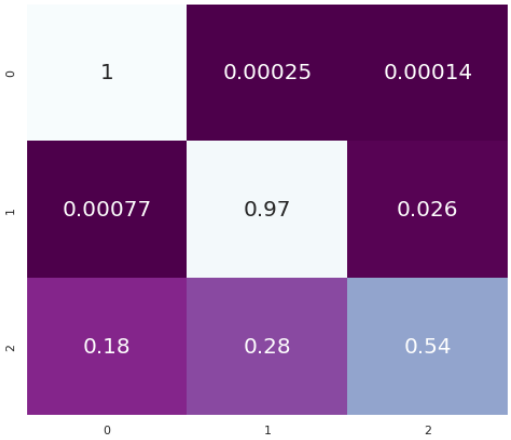
6. RESULTADOS

6.1 MÉTRICAS

En este segmento del documento, se detalla la comparación de los resultados dados por cada uno de los modelos que fueron implementados en el caso de estudio.

6.1.1 Matriz de confusión:

En el cuadro siguiente se aprecia el resultado dado para los cuatro modelos de clasificación que se utilizaron para el presente proyecto.

Regresión Logística	Árboles de decisión para clasificación
	
KNeighborsClassifier	Kmeans

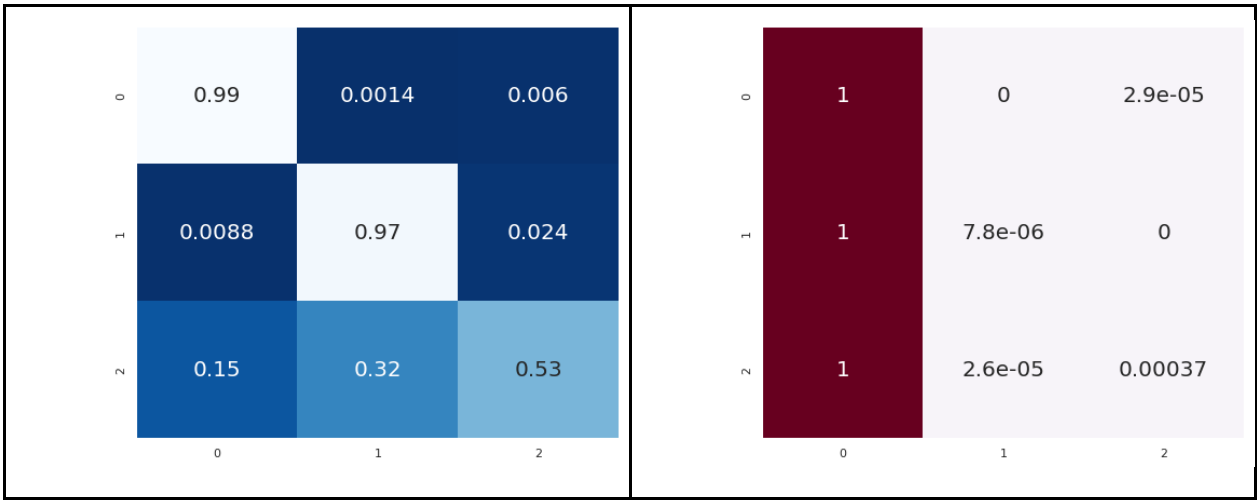


Tabla 16 Resultados de matrices de confusión para los modelos en estudio.

6.1.2 Accuracy y Balanced Accuracy:

En el siguiente cuadro se computaron los resultados obtenidos para las métricas accuracy y balanced_accuracy, en cada uno de los modelos elegidos.

Modelo \ Métrica	**Accuracy**	**Balanced Accuracy**
Regresión Logística	0.8507689096096305	0.6472423473195158
Árboles de decisión	0.9386264003896736	0.8415926748374027
KNeighborsClassifier	0.930530438768282	0.8300512869773048
Kmeans	0.5048532717951159	0.3336563583636372

Tabla 17 Valores resultantes accuracy y balanced_accuracy para los modelos en estudio

Finalmente se realiza un gráfico de barras, con el cual se comparan los valores resultantes correspondientes con las métricas en mención.

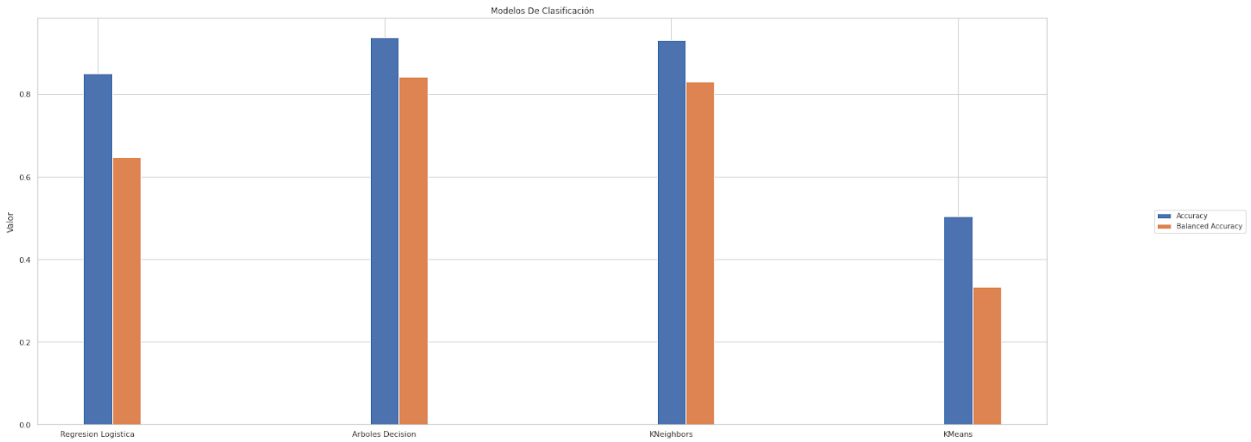


Figura 18 Gráfico comparativo resultado de métricas para modelos en estudio

7. CONCLUSIONES

Esta monografía comprende un trabajo de implementación de modelos analíticos para clasificación, abarcando desde la descripción de los datos hasta la evaluación de los modelos. Se realizó un análisis exploratorio para conocer el set de datos de una forma integral y de esta manera sacar el mejor provecho.

La implementación de varios modelos de clasificación, permitió validar que, a pesar de tener una única fuente de datos, las variaciones de parámetros y cálculos arrojan resultados con diferencias visibles, lo cual involucra al personal con conocimientos del negocio a interactuar con dichos resultados y partiendo de un análisis detallado poder tomar decisiones.

Los modelos de clasificación que tuvieron un mejor comportamiento, acorde al valor de accuracy, balanced accuracy y matriz de confusión, fueron los algoritmos correspondientes a árboles de decisión y KNeighbors.

Las métricas empleadas durante el proyecto fueron de gran utilidad, ya que facilitaron la evaluación, interpretación y comparación de los resultados entre los diferentes modelos implementados.

Para mejorar el proceso de análisis realizado durante la monográfica, se hace necesario tener la posibilidad de contactar con los dueños de la información. Dado que sería útil tener el conocimiento y descripción detallada de las variables que componen el set de datos, con lo cual se podría realizar un trabajo más profundo que se pudiera aplicar en la industria.

En un aspecto general para una implantación del análisis clasificatorio realizado en este trabajo, el conocimiento y visión del negocio daría una mayor claridad en la selección de las variables a analizar, con las cuales la utilidad del modelo aportaría buenos resultados a la entidad financiera.

8. BIBLIOGRAFIA

[1] «Datos Abiertos GOV.CO,» [En línea]. Available: https://www.datos.gov.co/Agricultura-y-Desarrollo-Rural/Colocaciones-de-Cr-dito-Sector-Agropecuario-2021-2/w3uf-w9ey. [Último acceso: 2022].

[2] V. M. Sebastian Raschka, Python Machine Learning, Marcombo, 2019.

[3] «UDEA / Especialización Analítica y Ciencia de Datos,» [En línea]. Available: https://github.com/UDEA-Esp-Analitica-y-Ciencia-de-Datos. [Último acceso: 2022].

[4] E. M. A. R. Rafael Caballero, Big Data Con Python, Alfaomega, 2019.