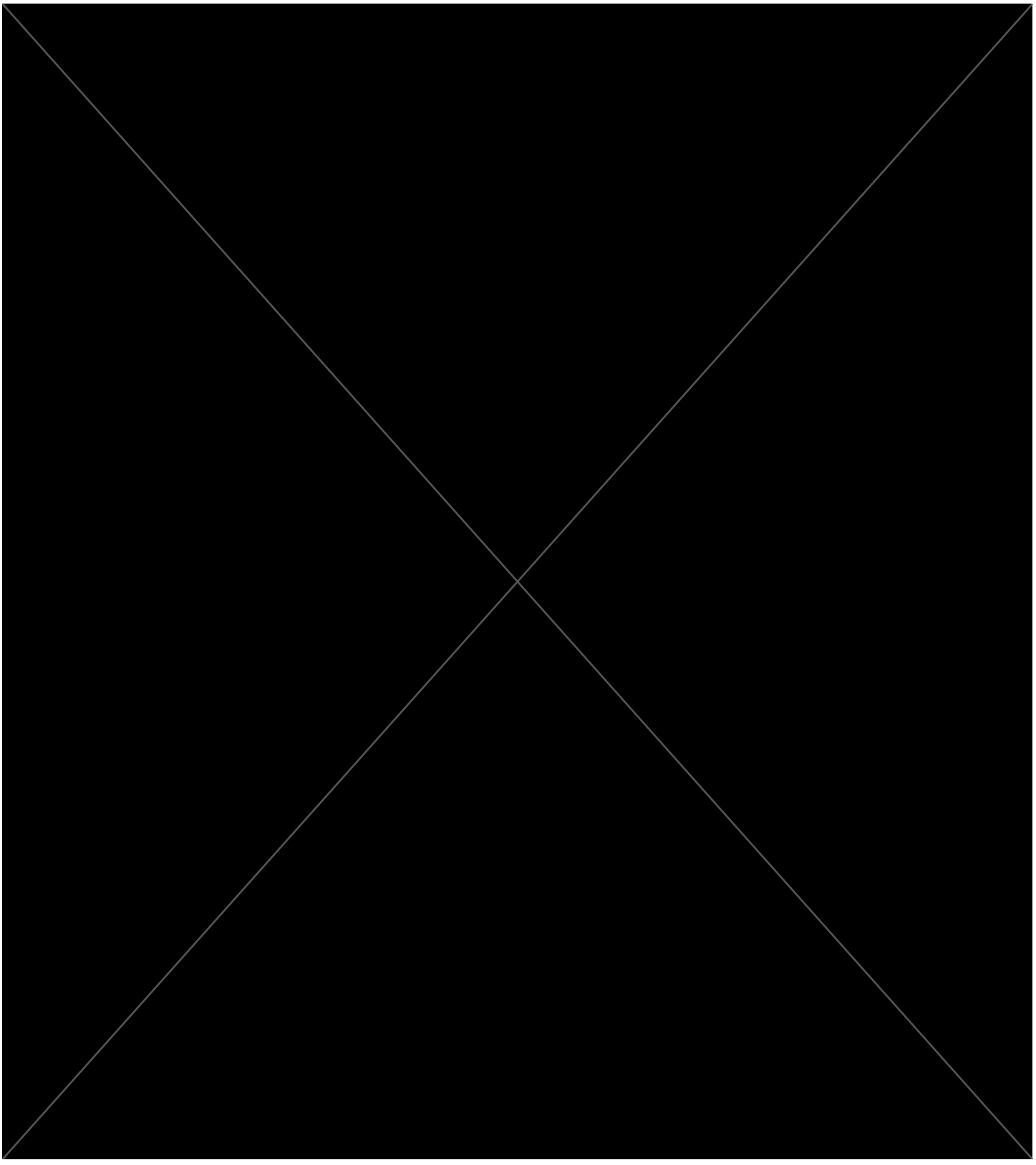


Marzo 2023

**Resumen**

En este proyecto se realizan modelos de predicción para conocer si un cliente será moroso en relación a distintas variables como el monto del préstamo, el plazo, entre otras. Esto nos ayuda a anticipar si algún cliente caerá en incumplimiento de pago y así tomar medidas al respecto.

# 1. Introducción

El término "calificación de crédito" se utiliza para describir el proceso de evaluar el riesgo que un cliente presenta de incumplir una obligación financiera (Kennedy K, 2013).

El objetivo es asignar a los clientes a uno de dos grupos: buenos y malos. Un miembro del grupo bueno se considera probable que cumpla su obligación financiera. Un miembro del grupo malo se considera probable que incumpla su obligación financiera. En su versión más simple, una tarjeta de calificación de crédito consiste en un conjunto de características que se utilizan para asignar una calificación de crédito a un cliente, indicando su nivel de riesgo. Esta calificación de crédito luego puede compararse con un umbral para tomar una decisión de préstamo. Ya que la calificación de crédito es esencialmente un problema de discriminación (bueno o malo), se puede recurrir a las numerosas técnicas de clasificación que se han sugerido en la literatura.

Por otro lado, la inteligencia artificial (IA) (McCarthy et al., 1955) es un campo de estudio que se basa en muchas disciplinas, incluyendo la ciencia de la computación, las matemáticas y la teoría de la información, la psicología cognitiva y la filosofía (Cook & Holder, 2001). El objetivo de la IA es desarrollar sistemas que proporcionen soluciones a tareas que tradicionalmente han sido consideradas como el dominio exclusivo de sistemas biológicos inteligentes. Como resultado de su naturaleza multidisciplinaria, los sistemas basados en IA son la manifestación de un amplio espectro de tecnologías y estrategias enfocadas en el desarrollo de: (i) modelos conceptuales; (ii) la representación formal de estos modelos; y (iii) estrategias de programación y hardware para implementar dichos modelos.

Dicho esto, se tomó para este proyecto una base de datos más de 67 mil datos de los cuales los más relevantes son los siguientes:

1. *Monto del Préstamo*: Es el monto del préstamo que dispuso el cliente.
2. *Tasa de Interés*: Tasa de interés del préstamo.
3. *Grado*: Es el nivel de riesgo asignado por el banco, siendo 1 el mejor y 7 el peor.
4. *Ocupación de Vivienda*: Es el nivel de ocupación de la vivienda del representante.
5. *Debito a Ingreso*: Proporción del pago total de la deuda mensual del representante dividido por el ingreso mensual autoinformado, excluyendo la hipoteca.
6. *Recuperansas*: Cargo posterior a la recuperación bruta.
7. *Saldo Actual Total*: Saldo total de todas las cuentas del representante.
8. *Estatus del Préstamo (Variable de Interés)*: Nos indica si el cliente es moroso (1) o no moroso (0).

Para lograr la predicción mas acertada nos enfocamos tanto en el apredizaje no supervisado y en el aprendizaje supervisado. El aprendizaje no supervisado es una técnica de aprendizaje automático en la que el modelo debe encontrar patrones y relaciones en los datos sin la ayuda de etiquetas o respuestas conocidas previamente (Bishop, 2006). En otras palabras, el modelo se encarga de descubrir información útil a partir de datos sin saber de antemano cuáles son los resultados esperados.

Por otro lado, el aprendizaje supervisado es un tipo de algoritmo de aprendizaje automático que utiliza datos etiquetados para entrenar un modelo y hacer predicciones o clasificaciones precisas (Alpaydin, 2010). En este tipo de aprendizaje, se proporcionan al algoritmo tanto los datos de entrada como las respuestas deseadas (o etiquetas) correspondientes a esas entradas. El objetivo del algoritmo es aprender una función que pueda mapear las entradas a las etiquetas de salida.

# 2. Marco Teoríco

El objetivo del aprendizaje automático es desarrollar herramientas y técnicas capaces de automatizar actividades humanas que consumen mucho tiempo de manera precisa y oportuna. Los sistemas basados en el aprendizaje automático logran este objetivo al intentar descubrir regularidades en un subconjunto de datos de entrenamiento, lo que permite la generación de hipótesis sobre todo el dominio de los datos. Como disciplina dentro de la inteligencia artificial, el rendimiento de un sistema basado en el aprendizaje automático debería mejorar a medida que adquiere experiencia o datos (Kennedy K, 2013).

Es por ello que para el entrenamiento de nuestro modelo se plantearon los siguientes algoritmos supervisados:

- Regresión Logística.
- Árbol de Decisión.
- Bosque Aleatorio.
- K Vecinos más Cercanos.

## 2.1. Regresión Logística

La Regresión Logística es un algoritmo de aprendizaje supervisado que se utiliza comúnmente en problemas de clasificación binaria. La regresión logística es quizás el algoritmo más utilizado dentro de la industria de puntuación de crédito al consumo (Hand & Zhou, 2009). El objetivo de la Regresión Logística es encontrar la relación entre un conjunto de variables predictoras (independientes) y una variable de resultado (dependiente) binaria, es decir, que toma valores en dos categorías posibles, como “sí/no”, “verdadero/falso” o “1/0”.

La Regresión Logística utiliza la función logística (también conocida como sigmoide) para modelar la probabilidad de que una instancia pertenezca a una categoría. La función logística tiene la siguiente forma matemática:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Donde $z$ es la suma ponderada de las variables predictoras:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$$

Aquí, $\beta_0$ es la intersección (también conocida como sesgo), $\beta_1$ a $\beta_n$ son los coeficientes de regresión y $x_1$ a $x_n$ son las variables predictoras.

La función logística toma cualquier valor de entrada y lo transforma en un valor entre 0 y 1, lo que lo convierte en una función adecuada para modelar la probabilidad. Si la probabilidad resultante es mayor que un umbral de decisión (por ejemplo, 0.5), se clasifica la instancia en la categoría positiva; de lo contrario, se clasifica en la categoría negativa.

El modelo de Regresión Logística se ajusta mediante la maximización de la función de verosimilitud logarítmica, que se define como la probabilidad de observar los datos dados los parámetros del modelo. La función de verosimilitud logarítmica es:

$$\ell(\beta) = \sum_{i=1}^{m} [y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i))]$$

Donde $m$ es el número de instancias, $y_i$ es la etiqueta real de la instancia $i$ y $z_i$ es la suma ponderada de las variables predictoras para la instancia $i$.

La optimización de la función de verosimilitud logarítmica se realiza mediante el algoritmo de descenso del gradiente, que ajusta iterativamente los coeficientes de regresión para minimizar la función de costo. La función de costo utilizada en la Regresión Logística es la Entropía Cruzada (Cross-Entropy), que mide la discrepancia entre las probabilidades predichas por el modelo y las etiquetas reales. La función de costo se define como:

$$\frac{\partial J(\beta)}{\partial \beta_j} = \frac{1}{m} \sum_{i=1}^{m} (\sigma(z_i) - y_i) x_{ij}$$

Donde $m$ es el número de instancias, $y_i$ es la etiqueta real de la instancia $i$ y $z_i$ es la suma ponderada de las variables predictoras para la instancia $i$.

El algoritmo de descenso del gradiente utiliza la derivada de la función de costo con respecto a los coeficientes de regresión. La función esta definida por:

$$\frac{\partial J(\beta)}{\partial \beta_j} = \frac{1}{m} \sum_{i=1}^{m} (\sigma(z_i) - y_i) x_{ij}$$

Y el algoritmo de descenso del gradiente:

$$\beta_j := \beta_j - \alpha \frac{\partial J(\beta)}{\partial \beta_j}$$

Anteriormente, una desventaja de la regresión logística era la intensidad computacional requerida durante la MLE, sin embargo, las mejoras en el hardware de computadoras han reducido esto como un problema. Una atracción de la regresión logística es que las características de entrada pueden ser continuas o discretas, o cualquier combinación de ambos tipos y no necesariamente tienen distribuciones normales (Lee, 2005).

## 2.2. Árbol de Decisión

El aprendizaje supervisado "Árbol de Decisión" es un algoritmo que se utiliza para la clasificación y regresión en problemas de aprendizaje automático. El objetivo de un árbol de decisión es crear un modelo que prediga el valor de una variable objetivo mediante la evaluación de una serie de reglas de decisión simples derivadas de las características de los datos. Los árboles de decisión se utilizan comúnmente en problemas de clasificación, pero también se pueden utilizar en problemas de regresión.

Un árbol de decisión se construye mediante la partición de los datos de entrenamiento en subconjuntos más pequeños y homogéneos basados en las características de los datos. Cada

partición se realiza a través de la evaluación de una característica en un nodo de decisión. La construcción del árbol continúa hasta que se alcanza un criterio de detención, como la profundidad máxima del árbol o el número mínimo de muestras requeridas en una hoja.

La construcción de un árbol de decisión se puede ver como la tarea de dividir el espacio de características en regiones cada vez más pequeñas y homogéneas, donde cada región corresponde a una hoja del árbol. La tarea de dividir el espacio de características se puede abordar mediante la minimización de una función de costo, que mide la heterogeneidad de las muestras en cada hoja del árbol.

La función de costo más comúnmente utilizada para los árboles de decisión es la entropía, que se define como:

$$H(T) = -\sum_{i=1}^{c} p(i|T) \log_2 p(i|T)$$

donde $T$ es un nodo del árbol, $c$ es el número de clases, y $p(i|T)$ es la proporción de muestras en el nodo $T$ que pertenecen a la clase $i$. La entropía mide la cantidad de incertidumbre o desorden en un conjunto de muestras.

El algoritmo de construcción del árbol de decisión selecciona la característica que minimiza la entropía después de la división, lo que se puede calcular mediante la ganancia de información, que se define como:

$$IG(D_p, f) = H(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} H(D_j)$$

donde $D_p$ es el conjunto de datos en el nodo padre, $f$ es la característica que se está evaluando, $m$ es el número de valores posibles de la característica $f$, $N_j$ es el número de muestras que pertenecen a la $j$-ésima rama, y $N_p$ es el número total de muestras en el nodo padre. La ganancia de información mide la reducción de la entropía después de la división, y la característica que maximiza la ganancia de información se selecciona como la característica de división.

El proceso de construcción del árbol se puede detener mediante la definición de un criterio de detención, como la profundidad máxima del árbol o el número mínimo de muestras requeridas para dividir un nodo. Una vez que se ha construido el árbol, se pueden utilizar diferentes estrategias de poda para reducir su complejidad y evitar el sobreajuste. La poda consiste en eliminar algunas de las ramas del árbol que no aportan información relevante o que son demasiado específicas para el conjunto de entrenamiento, lo que puede mejorar la capacidad de generalización del modelo.

## 2.3. Bosque Aleatorio

El aprendizaje supervisado de Bosque Aleatorio o "Random Forest" en ingles, es un algoritmo que se utiliza para la clasificación, regresión y otras tareas de aprendizaje automático. Es una técnica de ensamblado que combina múltiples árboles de decisión para producir un modelo más robusto y preciso.

El algoritmo de Bosque Aleatorio funciona creando múltiples árboles de decisión a partir de diferentes muestras de los datos de entrenamiento y características aleatorias, y combinando sus predicciones para producir una predicción final. Cada árbol de decisión se entrena en una muestra aleatoria de los datos de entrenamiento y utiliza una subconjunto aleatorio de las características para cada división de nodo. Esto aumenta la variabilidad entre los árboles individuales y reduce el riesgo de sobreajuste (Breiman 2001).

El proceso de entrenamiento del modelo se puede resumir en los siguientes pasos:

1. Seleccionar una muestra aleatoria de los datos de entrenamiento.

2. Seleccionar un subconjunto aleatorio de características para cada árbol.

3. Entrenar un árbol de decisión en la muestra de entrenamiento y características seleccionadas.

4. Repetir los pasos 1-3 un número predeterminado de veces para crear múltiples árboles de decisión.

5. Combinar las predicciones de los árboles para producir una predicción final.

La fórmula para la predicción en Bosque Aleatorio es la siguiente:

$$\hat{y} = \frac{1}{B} \sum_{i=1}^{B} h_i(\mathbf{x})$$

donde $\hat{y}$ es la predicción final, $n_{arboles}$ es el número de árboles en el bosque, $f_i(x)$ es la predicción del i-ésimo árbol y $x$ es el vector de características de entrada.

Una de las ventajas de Bosque Aleatorio es su capacidad para manejar grandes conjuntos de datos con muchas características. Además, puede manejar conjuntos de datos con valores perdidos y valores categóricos. También proporciona una medida de importancia de características, que se puede utilizar para seleccionar características relevantes para el modelo.

Siendo así, combina múltiples árboles de decisión para producir un modelo más preciso y robusto. Se utiliza comúnmente en problemas de clasificación, regresión y otras tareas de aprendizaje automático. Su capacidad para manejar grandes conjuntos de datos y características, así como para manejar valores perdidos y categóricos, lo convierte en una herramienta útil para una amplia variedad de aplicaciones.

## 2.4. K Vecinos Más Cercanos

El clasificador de vecinos más cercanos asigna una instancia según la clase de sus vecinos más cercanos. Se conoce más comúnmente como el vecino más cercano k (k-NN) ya que a menudo es más beneficioso considerar más de un vecino (ver Henley y Hand, 1996).

Esta sección describió ocho métodos de clasificación supervisada bien conocidos que son comúnmente utilizados por sistemas basados en aprendizaje automático para automatizar tareas como otorgar crédito o detectar fraudes. De hecho, debemos considerar el uso rutinario de dichos sistemas en la industria, educación y otros campos como la prueba definitiva para el aprendizaje automático (Langley y Simon, 1995). No es infrecuente en un entorno del mundo real que las observaciones de una clase en particular ocurran con mucha menos frecuencia en comparación con las poblaciones normales. Con bastante frecuencia, el costo de clasificar incorrectamente muestras de una clase rara es mayor que el caso contrario, por ejemplo, la clasificación de una transacción fraudulenta como una transacción legal (falso negativo). En el aprendizaje automático, un conjunto de datos con distribuciones de clase desiguales se puede considerar como un conjunto de datos desequilibrado.

El algoritmo se basa en la premisa de que los puntos de datos similares tienden a agruparse en el mismo espacio. Por lo tanto, se puede utilizar la cercanía en el espacio de características para determinar la clase a la que pertenece un punto de datos desconocido.

El algoritmo KNN se puede resumir en los siguientes pasos:

1. Calcular la distancia entre el punto de datos desconocido y todos los puntos de datos conocidos en el conjunto de entrenamiento.

2. Seleccionar los K puntos de datos más cercanos al punto desconocido.

3. Asignar la clase más común entre los K vecinos más cercanos al punto desconocido como la clase del punto desconocido.

La distancia entre dos puntos de datos se puede calcular utilizando diferentes métricas, pero la más común es la distancia euclidiana. La fórmula para la distancia euclidiana entre dos puntos A y B en un espacio de n dimensiones se puede expresar como:

$$d(A, B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2}$$

donde $A_i$ y $B_i$ son las coordenadas del punto A y B en la i-ésima dimensión.

Una vez que se han calculado las distancias, se seleccionan los K vecinos más cercanos al punto desconocido. Luego, la clase más común entre los K vecinos más cercanos se asigna al punto desconocido. En el caso de una clasificación binaria, la clase más común se puede determinar por mayoría de votos. En el caso de una clasificación múltiple, se puede utilizar el voto ponderado por la distancia para determinar la clase más probable.

Dónde la fórmula para el voto ponderado por la distancia en el caso de una clasificación múltiple:

$$\hat{y} = \arg\max_i \sum_{j=1}^{K} w_j I(y_j = i)$$

donde $\hat{y}$ es la clase predicha para el punto desconocido, $w_j$ es el peso del j-ésimo vecino más cercano, $y_j$ es la clase del j-ésimo vecino más cercano e $I(y_j = i)$ es una función indicadora que es 1 si la clase del j-ésimo vecino más cercano es i y 0 en caso contrario.

# 3. Metodología

Para cumplir con el propósito de este proyecto, el cual es predecir a través de ciertas características si una persona que obtiene un crédito será moroso o no, se tomaron en cuenta parámetros importantes como el monto del préstamo, la tasa de interés, el nivel de riesgo asignado por el banco, entre otras más.

Los datos utilizados fueron creados por "Machine Hack", plataforma en línea para competiciones de Aprendizaje Automatizado, la cual fue utilizada en el evento "Hackaton Deloitte ML Challenge" en diciembre de 2021. La base consta de un conjunto de datos de entrenamiento con más de 64 mil filas y 35 variables y otro conjunto de prueba con más de 28 mil datos y sus respectivas 35 variables. Dicho esto, nos concentraremos específicamente en 7 variables regresoras y 1 variable de interés:

1. *Monto del Préstamo*: Es el monto del préstamo que dispuso el cliente.

2. *Tasa de Interés*: Tasa de interés del préstamo.

3. *Grado*: Es el nivel de riesgo asignado por el banco, siendo 1 el mejor y 7 el peor.

4. *Ocupación de Vivienda*: Es el nivel de ocupación de la vivienda del representante.

5. *Debito a Ingreso*: Proporción del pago total de la deuda mensual del representante dividido por el ingreso mensual autoinformado, excluyendo la hipoteca.

6. *Recuperansas*: Cargo posterior a la recuperación bruta.

7. *Saldo Actual Total*: Saldo total de todas las cuentas del representante.

8. *Estatus del Préstamo (Variable de Interés)*: Nos indica si el cliente es moroso (1) o no moroso (0).

A continuación en la tabla 1 se muestran las primeras filas de nuestra base de datos de entrenamiento:

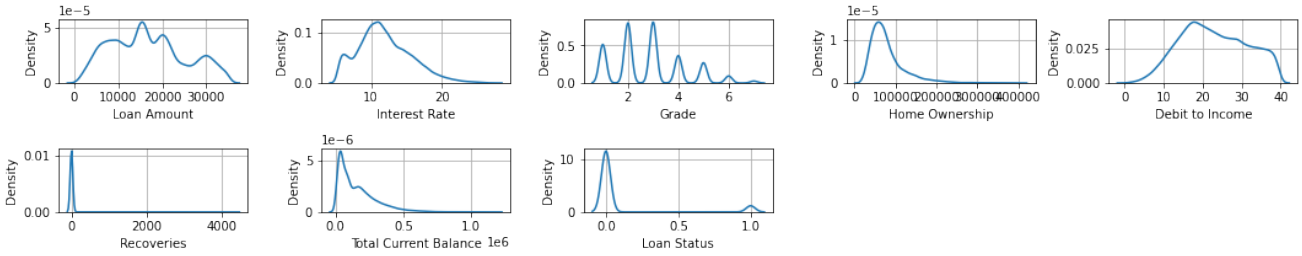
Tabla 1: Base de Entrenamiento para Morosidad en Clientes.

| Monto del Préstamo | Tasa de Interés | Grado | Ocupación de Vivienda | Debito a Ingreso | Recuperansas | Saldo Actual Total | Estatus del Préstamo |
|---|---|---|---|---|---|---|---|
| 10000 | 11.135007 | 2 | 176346.6267 | 16.284758 | 2.498291 | 311301 | 0 |
| 3609 | 12.237563 | 3 | 39833.921 | 15.412409 | 2.377215 | 182610 | 0 |
| 28276 | 12.545884 | 6 | 91506.69105 | 28.137619 | 4.316277 | 89801 | 0 |
| 11170 | 16.731201 | 3 | 108286.5759 | 18.04373 | 0.10702 | 9189 | 0 |
| 16890 | 15.0083 | 3 | 44234.82545 | 17.209886 | 1294.818751 | 126029 | 0 |

Para todo el proyecto se hizo uso de la herramienta Python, el cual es un lenguaje de programación para su uso estadístico.

Antes de realizar el aprendizaje supervisado se analizó de cada variable su distribución para así hacer un preprocesamiento correcto e ir ajustando las variables para realizar el modelo. Donde "Loan Amount" es Monto del Préstamo; "Interest Rate" es Tasa de Interés; "Grade" es Grado; "Home Ownership" es Ocupación de Vivienda; "Debit to Income" es Debito a Ingresos; "Recoveries" es Recuperansas; "Total Current Balance" es Saldo Actual Total; "Loan Status" es Estatus del Préstamo.

Figura 1: Densidad de Cada Variable.




De lo anterior se puede observar haciendo suposiciones que las variables de la tasa de interés, la ocupación de la vivienda y el saldo total actual tienden a seguir una función de densidad logarítmica normal. Otras variables como el monto del préstamo y el débito a ingresos se podría decir que los datos se distribuyen normal.

Dicho esto se estandarizarán las variables para poder tener un mejor resultado en los modelos por aplicar dentro del aprendizaje supervisado. A su vez, se retira la variable de interes "Estatus del Préstamo" para continuar con la creación de los modelos.

Tabla 2: Datos Estandarizados.

| Monto del Préstamo | Tasa de Interés | Grado | Ocupación de Vivienda | Debito a Ingreso | Recuperansas | Saldo Actual Total |
|---|---|---|---|---|---|---|
| -0.494173 | -0.067255 | -0.5 | 2.48036 | -0.453543 | -0.13725 | 1.098588 |
| -1.02524 | 0.153106 | 0 | -0.680332 | -0.51936 | -0.167486 | 0.375906 |
| 1.024493 | 0.214728 | 1.5 | 0.516053 | 0.440736 | 0.316751 | -0.145277 |
| -0.39695 | 1.051221 | 0 | 0.904559 | -0.320831 | -0.734415 | -0.597965 |
| 0.07836 | 0.706875 | 0 | -0.578438 | -0.383743 | 322.59029 | 0.058167 |

Planteado lo anterior, se procedió a realizar la creación de los modelos donde la metodología inicial fue hacer iteraciones con la regresión logística, árbol de decisión, bosque aleatorio y k vecinos más cercanos. Para ello se hace el ajuste del modelo de acuerdo a los datos de entrenamiento, se procede a predecir la variable de interés "Estatus del Préstamo" en función a las regresoras de la base de entrenamiento, empieza a predecir el "Estatus del Préstamo" de la base de prueba de acuerdo a las variables regresoras de esta misma base y por último mostramos la precisión de cada uno de los modelos, comparandolos a su vez con las matrices de confusión para ver cuantos datos pudieron predecir correctamente cada uno de los modelos. A continuación se muestra el procedimiento utilizado en Python para la creación de este método.

```
lr = LogisticRegression()
dt = DecisionTreeClassifier()
rf = RandomForestClassifier()
knn = KNeighborsClassifier()

model_list = [lr,dt,rf,knn,gnb]
cm = []
train_acc = []
for i in model_list:
    i_model = i.fit(xtrain,ytrain)
    ypred_train = i_model.predict(xtrain)
    ypred_test = i_model.predict(xtest)
    train_acc.append(accuracy_score(ytrain,ypred_train))
```

Listing 1: Procedimiento de Ajuste y Predicción por Modelo

Más adelante en la sección de resultados se muestran en la tabla 4 la precisión de este método, del cual se percató que podría sufrir un sobre ajuste en los modelos y se procedió a realizar un "SMOTE" sobre los datos, esta es una técnica de sobre muestreo que generará muestras sintéticas de la clase minoritaria (morosos) para equilibrar la distribución de clases.

Este método funciona seleccionando una muestra de la clase minoritaria y encontrando sus k-vecinos más cercanos para seleccionar uno de ellos al azar e interpolar una nueva muestra entre la muestra original y el vecino elegido, creando sintéticamente una nueva muestra de la clase minoritaria y repitiendo este proceso hasta que tengamos equilibrio entre ambas clases.

Este proceso se realiza porque el desequilibrio de clases puede hacer que los clasificadores se sesguen hacia la clase mayoritaria y les resulte difícil identificar correctamente los ejemplos de la clase minoritaria, lo que conduce a un rendimiento deficiente y puntuaciones de recuperación bajas.

La fórmula utilizada para la generación de nuevas instancias a través de SMOTE es la siguiente:

$$X_{nueva} = X_i + \lambda(X_{zi} - X_i)$$

Antes de realizar este procedimiento, se contaban con 48,941 personas no morosas y 5,029 personas morosas de la base de entrenamiento. Después de realizar el método "Smote" se obtuvo lo siguiente:

Tabla 3: Datos de Entrenamiento con SMOTE.

| Estatus del Préstamo | Observaciones Sin SMOTE | Observaciones Con SMOTE |
|---|---|---|
| No Moroso | 48,941 | 48,941 |
| Moroso | 5,029 | 48,941 |

Realizado este SMOTE a los datos se procedio a realizar de nuevo el ajuste del modelo y la predicción de nuestra variable de interés.

# 4. Resultados

Planteada la metodología anterior en nuestro primer método donde se realiza el ajuste a partir de los datos de entrenamiento donde se observaron las siguientes precisiones para cada modelo.

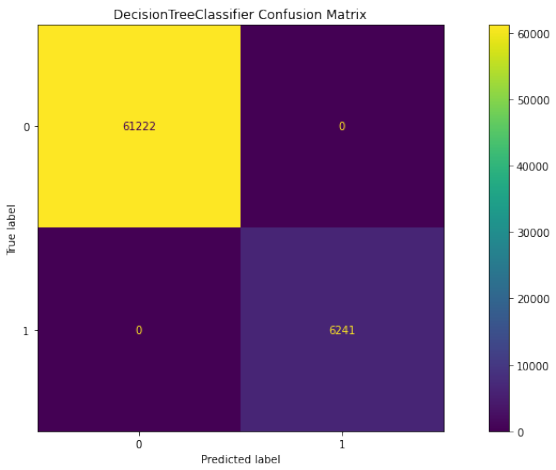
Tabla 4: Precisión de los Modelos.

| Modelo | Precisión |
|---|---|
| Regresión Logística. | 90.75 % |
| Árbol de Decisión. | 100.00 % |
| Bosque Aleatorio. | 100.00 % |
| K Vecinos más Cercanos. | 90.85 % |

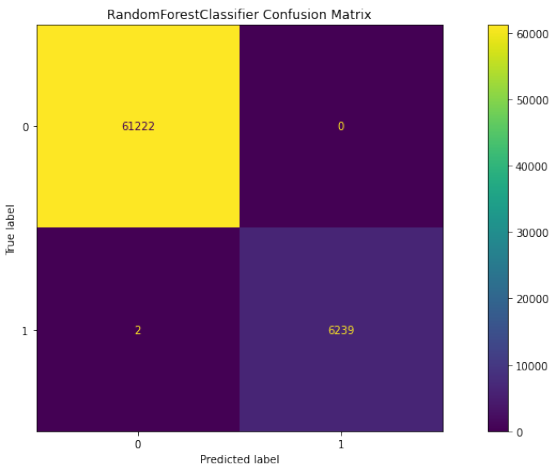
Para este método la precisión se obtiene comparando la variable de interés predecida de entrenamiento contra la verdadera de esa misma base de datos. Es por ello que probablemente se ejecutó con un sesgo en la ejecución del mismo. A continuación mostramos las matrices de confusión de los 2 mejores modelos para comprobar lo mencionado.

Figura 2: Matriz de Confusión.

(a) Árbol de Decisión




(b) Bosque Aleatorio




De la tabla 4 y la figura 2 podemos ver que los modelos nos arrojaron ajustes muy buenos, en el modelo de árbol de decisión obtuvimos todos los resultados bien predecidos. Por otra parte en el modelo de bosque aleatorio se observan que unicamente 2 observaciones de 6,241 morosos fueron incorrectas.

Sin embargo, por su alto nivel de precisión se optó por mejor partir la base de entrenamiento en 80 % para entrenamiento y el 20 % para la prueba. Esto para comparar directamente con una base con resultados ya preestablecidos. A su vez, realizamos el procedimiento de "SMOTE" en los datos para que los clasificadores no tengan un sesgo hacia la clase mayoritaria.

Los resultados de precisión para los modelos fueron los siguientes:

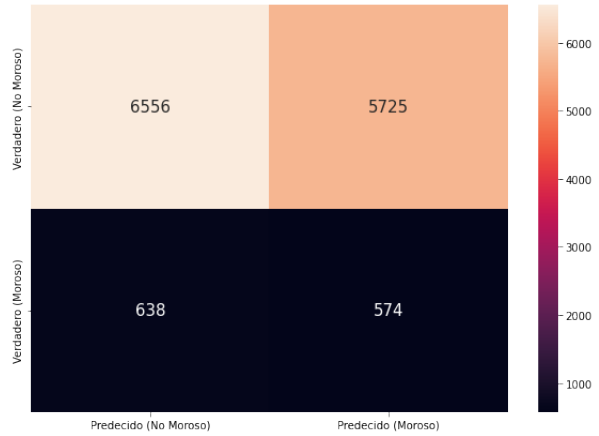
Tabla 5: Precisión de los Modelos.

| Modelo | Precisión |
|---|---|
| Regresión Logística. | 47.36 % |
| Árbol de Decisión. | 17.41 % |
| Bosque Aleatorio. | 3.55 % |
| K Vecinos más Cercanos. | 8.72 % |

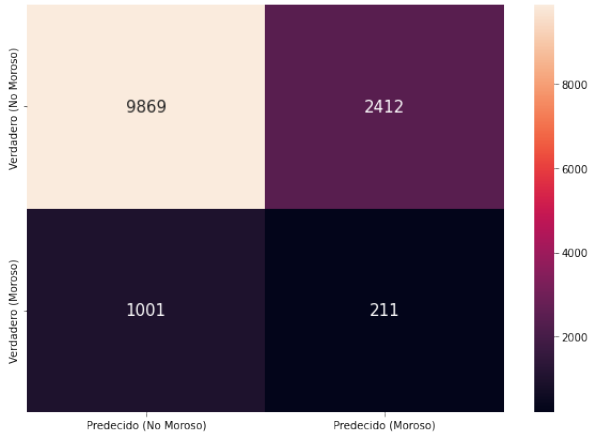
Se puede observar de la tabla 5 que nuestro mejor resultado llega a un 47.36 % utilizando nuestro modelo de regresión logística. A continuación mostraemos en la figura 3 especificamento cuántos verdaderos positivos obtuvimos en cada uno de los modelos.

Figura 3: Matriz de Confusión con SOMTE.

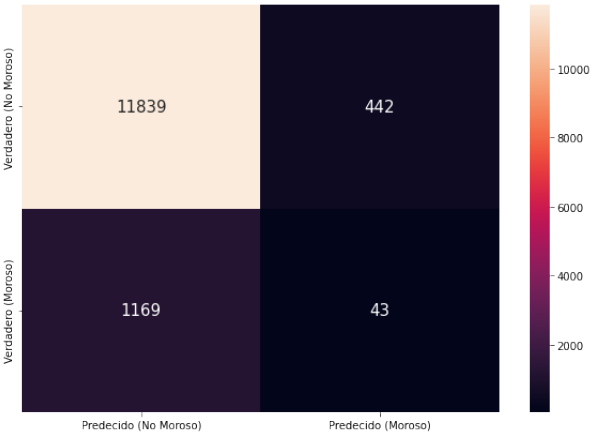
(a) Regresión Logística



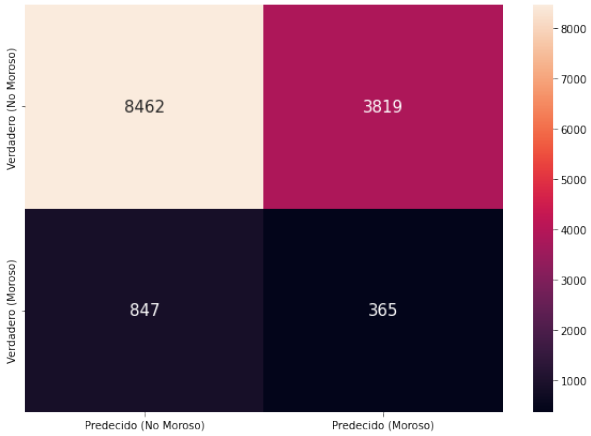

(b) Árbol de Decisión




(c) Bosque Aleatorio




(d) K Vecinos más Cercanos




De lo anterior podemos observar que para la regresión logística obtuvimos 574 resultados ciertos para la predicción de morosos contra 1,212 casos. Por otro lado obtuvimos 6,556 para la predicción de no morosos contra 12,281. En cambio los demás modelos solo nos arrojaron resultados de precisión por debajo del 20 %.

# 5. Conclusiones

Del estudio anterior podemos concluir lo siguiente:

- Que se tendría que revalorar la importancia de las variables para tomar en cuenta otras de las que fueron eliminadas para la elaboración del proyecto. Esto incluyendo las variables categóricas convirtiéndolas a través de procesos estadísticos en información útil para el desenlace del proyecto.
- En la creación de los métodos se observó que al predecir la misma base de entrenamiento con el modelo de árbol de decisión el modelo de bosque aleatorio se obtuvo una precisión del 100 %, sin embargo al tratar de replicarlo a los datos de prueba, estos no resultaron ser precisos. Por lo que fue necesario aplicar diferentes técnicas en los mismos para tratar de ajustar los modelos y que nos dieran mejor una precisión de predicción.
- Al momento de realizar el método con SMOTE se obtuvieron mejores resultados, donde el mejor modelo de predicción fue el de Regresión Logística con un nivel de precisión del 47.36 %.

Dicho esto será de gran importancia realizar distintas pruebas haciendo una mejor categorización de los datos. Probablemente realizando procedimientos de caracterización aritmética sobre los datos tanto numéricos como categóricos se puedan llegar a mejores resultados.

Por otro lado, es importante mencionar que la calidad de los datos utilizados para la elaboración de los modelos de predicción es esencial para obtener resultados precisos. Es necesario asegurarse de que los datos sean completos, relevantes y de calidad. Por lo que sería de gran relevancia revisar bases de datos de fuentes más seguras y relevantes.

El trabajo futuro debería concentrarse en comparar el rendimiento de los modelos de clasificación con modelos de duración, como el análisis de supervivencia. Además, hoy en día con la tecnología y el gran avance de la banca digital, será necesario implementar mejores técnicas para la predicción de impago por parte de los clientes, esto debido a principalmente que este tipo de instituciones tienen por lo general un gran crecimiento en materia de crédito, pero no tienen una recuperación pronto del mismo. En varias instituciones de crédito "pequeñas" se puede observar que año tras año incrementa su cartera vencida y no tienen índices altos de recuperación. Justo por esto, es importante incorporar este tipo de modelos en estas instituciones para que así puedan tener una mejor toma de decisiones.

# 6. Referencias

Referencias:

Kennedy, K. (2013). Credit scoring using machine learning. Doctoral thesis. Technological University Dublin. doi:10.21427/D7NC7J.

McCarthy, Y. & McQuinn, K. (2010). How are Irish households coping with their mortgage repayments Information from the SILC Survey. Research Technical Papers, Central Bank & Financial Services Authority of Ireland (CBFSAI),
available at http://www.centralbank.ie/publications/documents/2RT10.pdf, last accessed 29 January 2013. 210, 213

Cook, D. & Holder, L. (2001). A client-server interactive tool for integrated artificial intelligence curriculum. In Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference AAAI Press. 2

Alpaydin, E. (2010). Introduction to machine learning (2da ed.). Cambridge, MA: MIT Press.

Bishop, C. M. (2006). Pattern recognition and machine learning. New York: Springer.

Hand, D. & Zhou, F. (2009). Evaluating models for classifying customers in retail banking collections. Journal of the Operational Research Society, 61, 1540–1547. 23, 91, 124, 180

Lee, S. (2005). Application of logistic regression model and its validation for landslide susceptibility mapping using gis and remote sensing data. International Journal of Remote Sensing, 26, 1477–1491. 25

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Henley, W. & Hand, D. (1996). A k-nearest-neighbour classifier for assessing consumer credit risk. The Statistician, 45, 77–95. 28

Langley, P. & Simon, H. (1995). Applications of machine learning and rule induction. Communications of the ACM , 38, 54–64. 28