UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO
FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y TELECOMUNICACIONES

WEKA

INTEGRANTES:

- Camino Puma Ronald
- Torrez Vaca Andres
- Vino Apaza Vanesa

MATERIA: Sistemas para el Soporte y la Toma de Decisiones

SIGLA: INF-432 “SA”

DOCENTE: Ing. Miguel Peinado Pereira

Santa Cruz – Bolivia
2024

Índice

1. ¿Qué pasos se siguieron para llegar a esos datos?

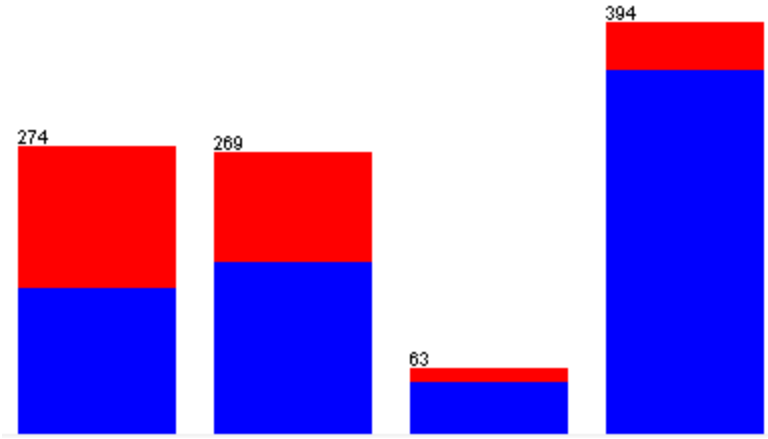
- **Selección: Identificación de fuentes y datos relevantes.**
  Proceso de identificación de datos pudo comenzar con la recopilación de datos relacionados con clientes y sus historiales crediticios. Las fuentes pueden incluir:
    - Bases de datos internas del banco sobre, cuentas, historial de pago, incumplimiento, ingresos, etc.
    - Registro de agencias externas información crediticia y financieras obtenida de entidades especializadas en análisis de crédito.
    - Formularios de solicitud de crédito con datos proporcionados directamente por los clientes al solicitar servicios financieros, como el monto del préstamo solicitado o referencias laborales.

- **Preprocesamiento: Limpieza y preparación de los datos.**
  Pasos posibles realizados para el análisis de datos:
    - **Limpieza de datos.** Eliminar datos duplicados, gestionar valores nulos y corregir inconsistencias. Clientes sin datos importantes como credit_amount o credit_status podrían ser descartados.
    - **Eliminación de valores inconsistentes o faltantes.** Fila con valores poco claros como **no checking** podrían haberse manejado como categoría explicitas como "**Sin comprobar**".
      Valores incompletos o irrelevantes en **savings_status** como **no known saving** pueden haber sido categorizados como "**sin ahorros**".
    - **Manejo de valores externos.** En **credit_amount** valores que se alejan significativamente de los rangos típicos (250 - 18424) pueden ser descartados o eliminados.
    - **Revisión de inconsistencia.** Datos como **employment** sean coherentes, un cliente **unemployment** (desempleado) probablemente no debería solicitar créditos grandes.

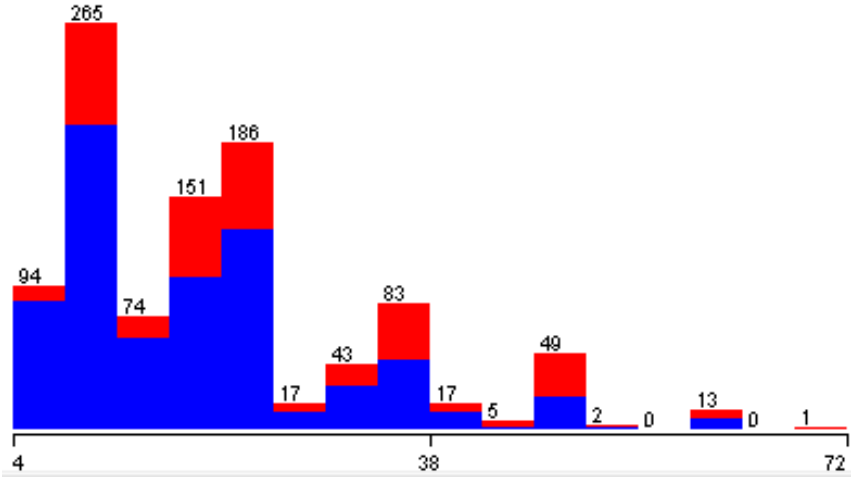
- **Transformación: Ajuste y generación de nuevas características.**
    - **Agrupación de valores. savings_status** (estados de ahorro) se agrupa en categoría (<100, 100<=X<500, >=1000) para la captura de datos o patrones fácilmente.
    - **Creación de etiquetas de clasificación. class** se etiqueto explícitamente como **Good** o **Bad**, para los análisis supervisados.
    - **Normalización de atributos continuos. credit_amount** normalizado para evitar sesgos. **Age** reescalado en rangos adecuados para igualar su importancia con otros atributos.
    - **Generación de nuevas características.** Se podría crear un atributo adicional como **credit_risk** basado en una combinación de **credit_history** y **savings_status.**
      El atributo **employment** podría haberse codificado numéricamente.

2. ¿Qué significa estos datos?

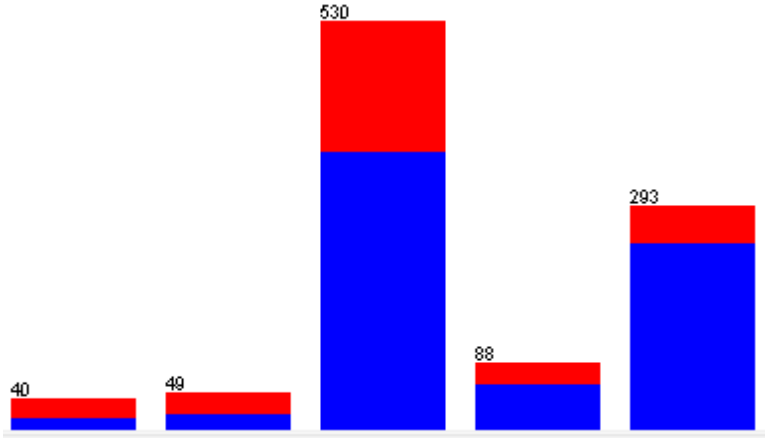
- **checking_status (estado de la cuenta).** Representa la situación financiera con respecto a una cuenta. Saldo negativo, <200, >=200, sin saldo.



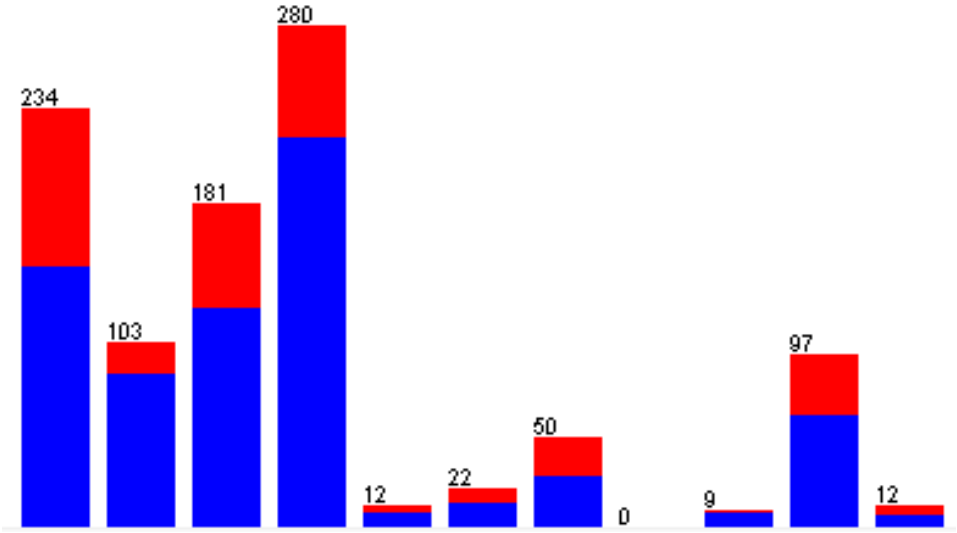
- **Duration (duración).** Duración del crédito solicitado en meses, créditos mas largos pueden implicar mayor riesgo financiero 4 a 72 meses.



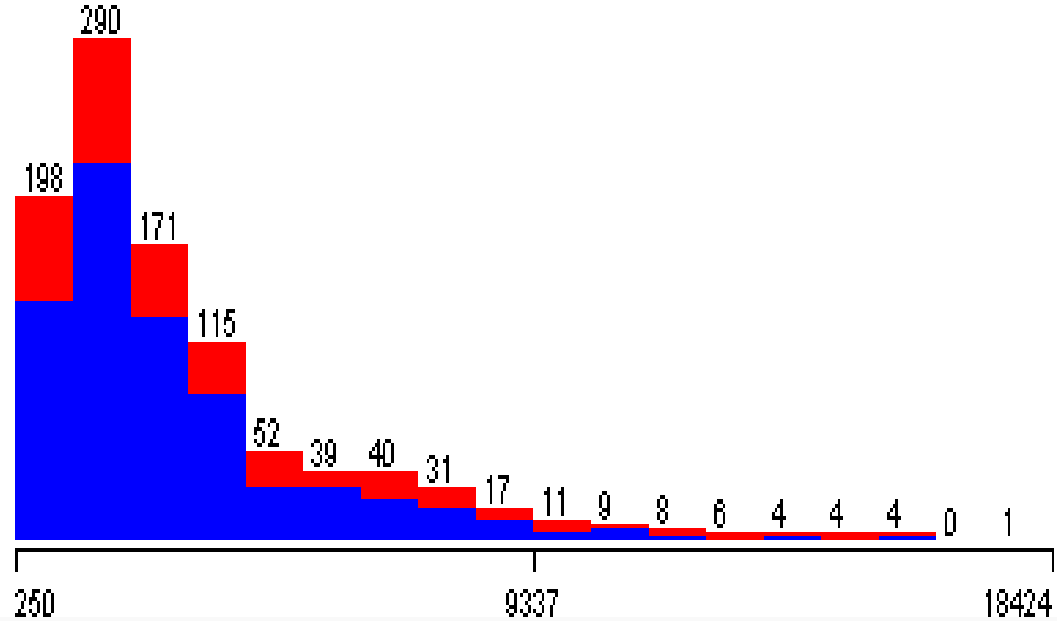
- **Credit_history (historal de credito).** Historial de crédito del cliente, refleja el estado de los créditos previos todo pagado, pago existente, retrasado, previamente, critico.



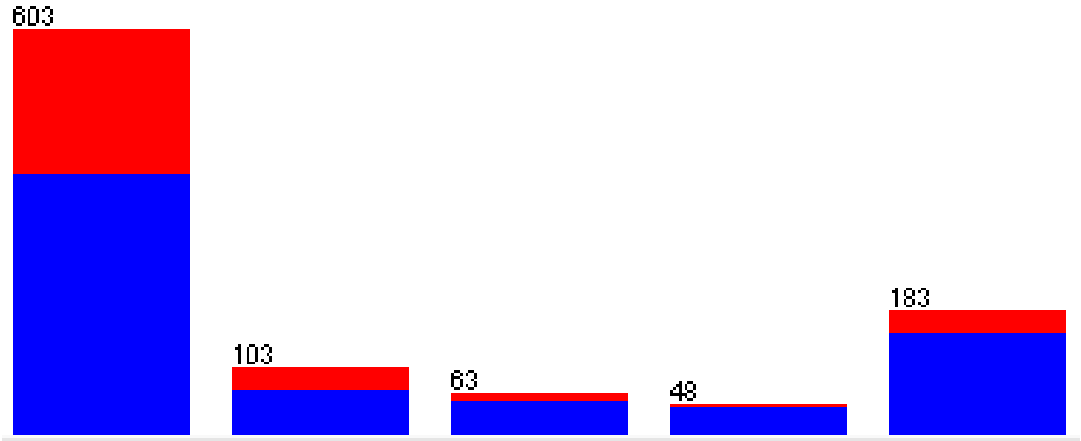
- **Purpose.** Propósito del crédito solicitado. Propósito de inversión menor riesgo nuevo auto, radio, educación, vacación, negocio.



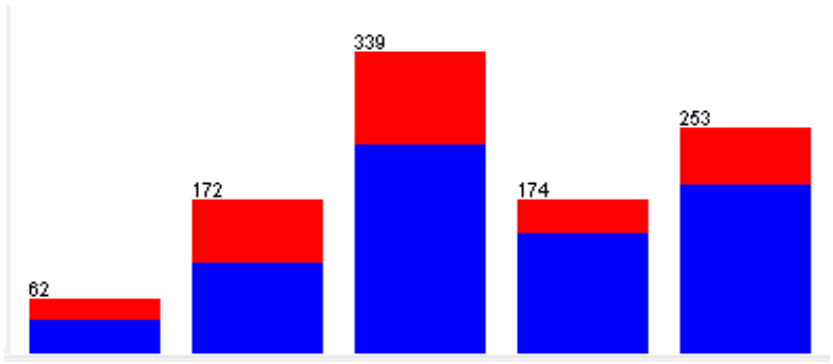
- **Credit_amount.** Monto de crédito solicitado, Nivel de deuda, montos altos. Valores de 250 0 18,424 unidades monetarias.



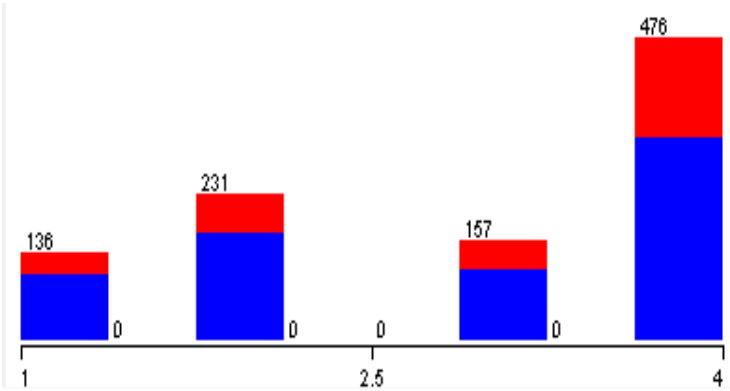
- **Savings_status.** Estado de los ahorros, capacidad de respaldo financiero. Valores: sin ahorros, <100, 100-500, 500-1000, >=1000



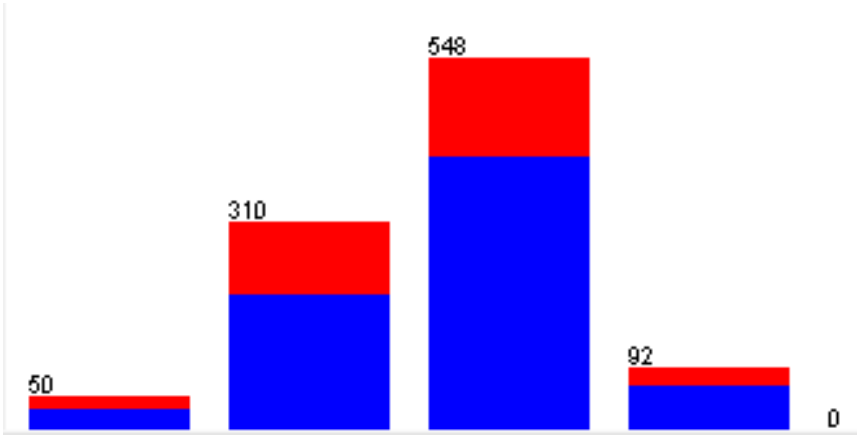
- **Employment.** Tiempo de empleo del cliente. Mayor antigüedad menor riesgo, capacidad de pago. Calores: <1 año, 1-4 años, 4-7 años, >=7 años, desempleado.



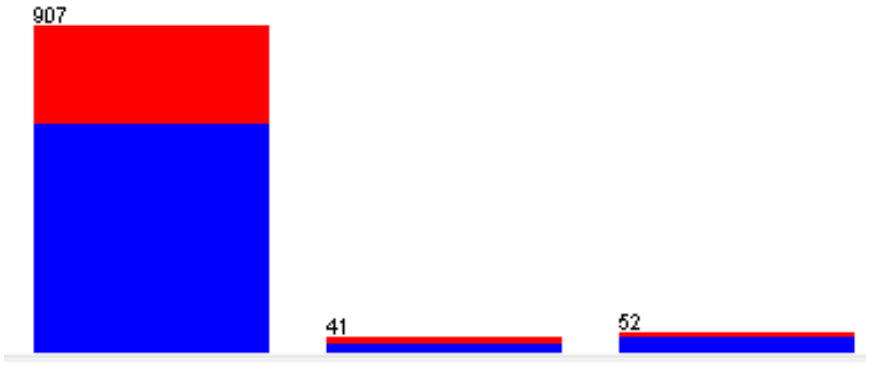
- **Installmente_commitment.** Porcentaje del ingreso mensual destinado al pago del crédito. Valores: 1% s 5%.



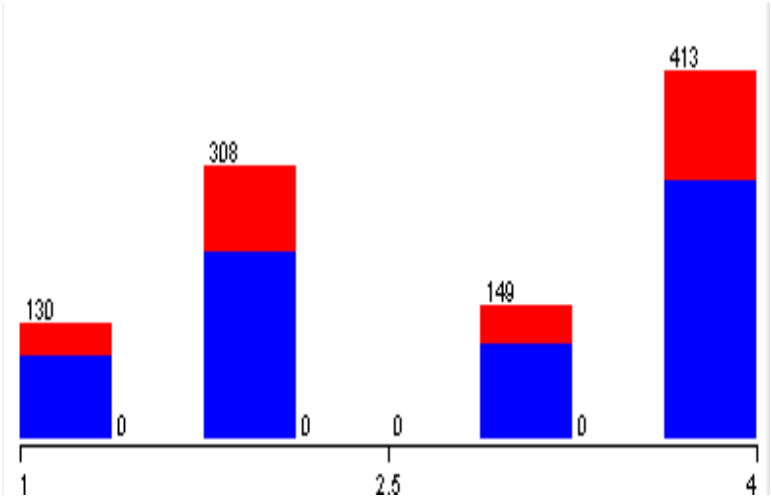
- **Personal_status.** Estado civil y género. Pueden correlacionar con el riesgo. Valores: casado, soltero, divorciado.



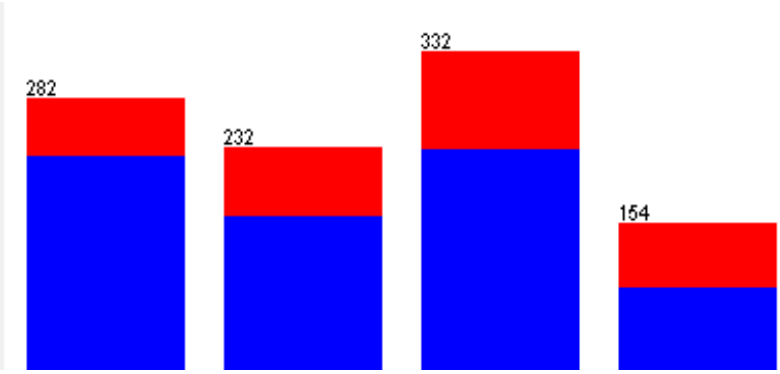
- **Other_parties.** Otras partes responsables asociadas al crédito. Valores: ninguno, codeudor, fidor,



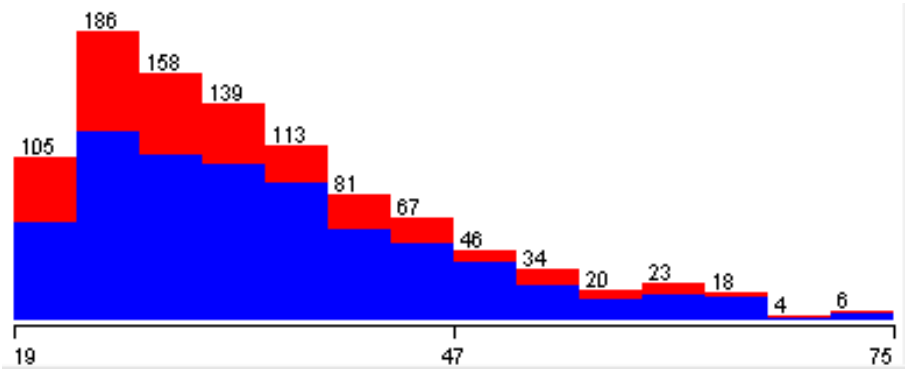
- **Residence_since.** Tiempo de residencia en años, mayor residencia mayor estabilidad. Valor: 1 a 4 años.



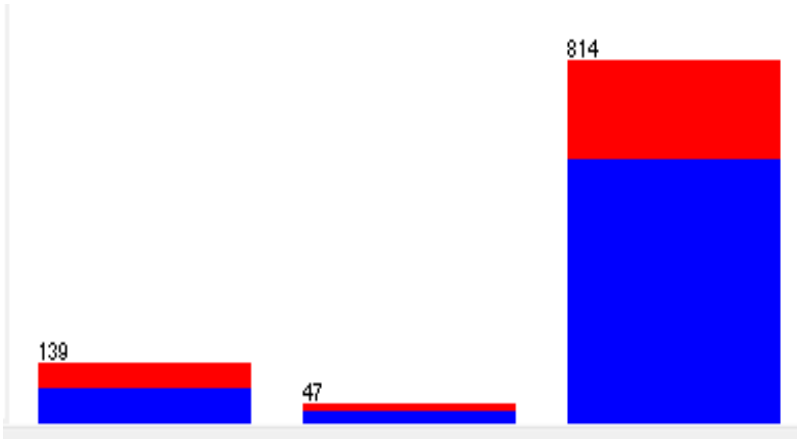
- **Property_magnitude.** Tipo de propiedad del cliente. Es garantía en caso de incumplimiento, Inmuebles, automóvil, Otros archivos, sin propiedad.



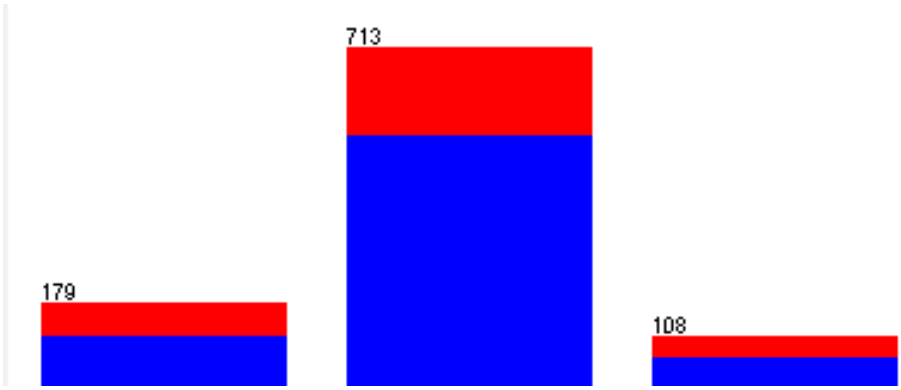
- **Age.** Edad del cliente, influye en la evaluación del riesgo y capacidad de pago. Valores: 18 a 75 años.



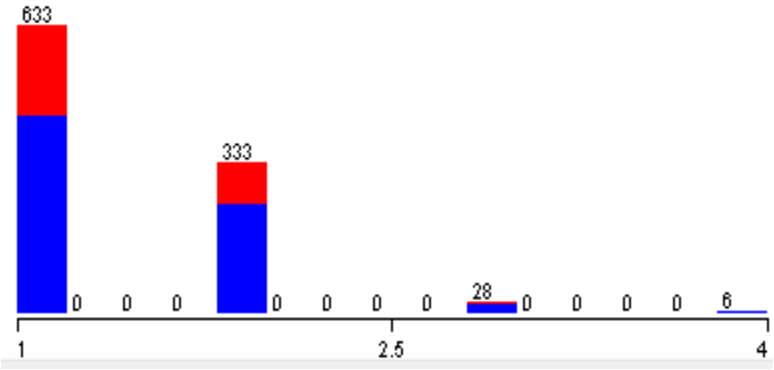
- **Other_pyment_plans.** Otros planes de pago, otros compromisos mayor riesgo. Valores: ninguno, blanco, tiendas.



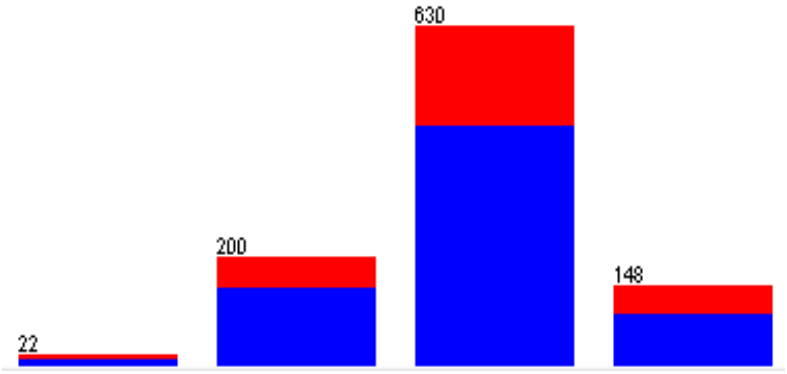
- Housing. Tipo de vivienda, estabilidad y posibles garantías. Valores: propia, alquilada, gratuita.



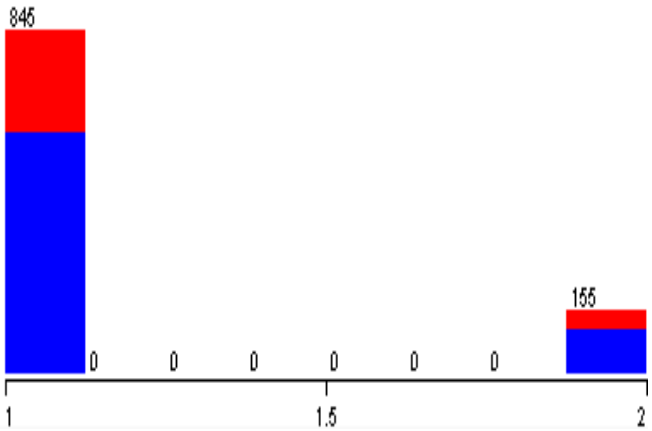
- **Existing_credits.** Número de créditos activos más créditos mayor riesgo 1 a 4 créditos.



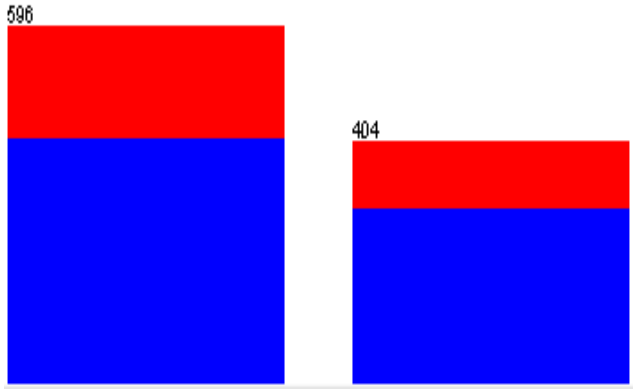
- **Job.** Tipo de empleo estabilidad e ingresos potenciales. Valores: desempleado, no calificado, calificado, autónomo/altos ingresos.



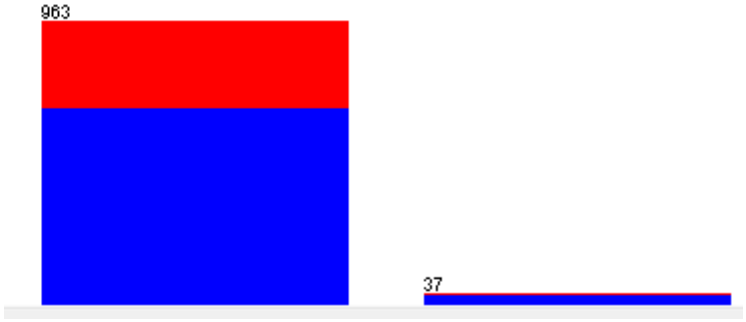
- **Num_dependents.** Número de personas dependientes mayor numero implica mayores responsabilidades financieras. Valores: 1 a 2 dependientes.



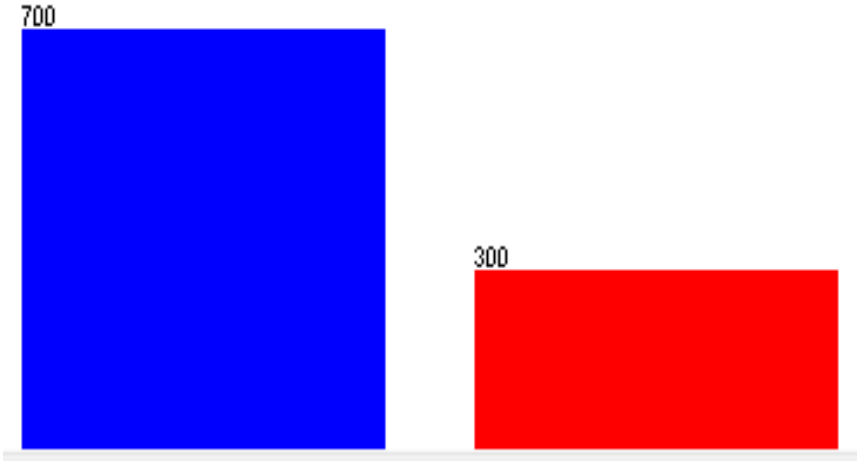
- **Own_telephone.** Telefono propio mayor estabilidad. Valores: si, no.



- **Foreing_worker.** Trabajador extranjero evaluación del riesgo según políticas locales. Valores: si, no.



- **Class.** Clasificacion del cliente determina si el cliente es apto para el crédito. Valores: Good (buen historial), bad (historial negativo).



3. ¿Cuál es el propósito que se quiere alcanzar del conjunto de datos?

- **Evaluar la Solvencia Finaciera:**
  - Atributos como checking_status, savings_status y credit_amount proporcionan una visión directa de la situación financiera del cliente, incluyendo el saldo de la cuenta, nivel de ahorros y el monto de crédito solicitado.
  - **Propósito:** Relacionar estas variables permite entender si un cliente con bajos ahorros y saldos negativos tiene un monto de crédito solicitado acorde a su capacidad financiera.

- **Estimar la Capacidad de Pago:**
  - La combinación de **employment, installment_commitment** y **income_lavel** ayuda a determinar si un cliente tiene una fuente de ingresos estable y puede destinar un porcentaje adecuado al pago de sus compromisos crediticios
  - **Propósito:** Identificar clientes con alta probabilidad de cumplimiento al analizar la antigüedad laboral junto con la proporción de ingresos destinada al crédito.

- **Segmentar Riesgo por Historial Crediticio:**
  - **Credit_history** y **existing_credits** son indicadores clave del comportamiento pasado del cliente. El historial de pago exitosos o retrasados junto con el número de crédito activos refleja la responsabilidad financiera.
  - **Propósito:** Predecir la probabilidad de incumplimiento futuro al analizar el historial de pagos combinado con la carga crediticia actual.

- **Identificar Estabilidad Residencial y Garantías:**
  - **residence_since** y **property_magnitude** ofrecen información sobre la estabilidad del cliente y las posibles garantías para respaldar el crédito.
  - **Propósito:** Relacionar el tiempo de residencia y las propiedades disponibles permite evaluar el nivel de estabilidad y la capacidad de ofrecer garantías en caso de incumplimiento.

- **Evaluar Factores Demograficos y Sociales:**
  - Variables como **age, personal_status** y **num_dependents** aportan contexto sobre responsabilidades financieras del cliente y su estabilidad social.
  - **Propósito:** Relacionar estas características ayuda a determinar el nivel de riesgo según el perfil demográfico (por ejemplo, clientes jóvenes son dependientes y pueden ser de mayor riesgo)

- **Detectar Factores de Riengo Externos:**
  - **Other_parties** y **foreingn_worker** representan factores externos que pueden influir en el riesgo crediticio, como la dependencia de un codeudor o las políticas locales hacia trabajadores extranjeros.
  - **Propósito:** Evaluar como estos factores externos afectan la solvencia y estabilidad financiera de los clientes.

- **Optimizacion de Modelos Predictivos:**
    - Al normalizar y agrupar los atributos continuos como **duration, credit_amount** y **age,** se busca reducir la complejidad y mejorar la capacidad predictiva de los modelos.
    - **Propósito:** Desarrollar un modelo que clasifique de manera precisa a los clientes en **Good** o **Bad** para optimizar las decisiones de otorgamiento de crédito.

4. Algoritmos Aplicados.

FUNCTIONS LOGISTIC.

```
=== Run information ===
Scheme:       weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Relation:     german_credit
Instances:    1000
Attributes:   21
              checking_status
              duration
              credit_history
              purpose
              credit_amount
              savings_status
              employment
              installment_commitment
              personal_status
              other_parties
              residence_since
              property_magnitude
              age
              other_payment_plans
              housing
              existing_credits
              job
              num_dependents
              own_telephone
              foreign_worker
              class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===
Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                                   Class
Variable                             good
=========================================================
checking_status=<0                 -0.778
checking_status=0<=X<200           -0.4032
checking_status=>=200               0.1877
checking_status=no checking         0.9339
duration                           -0.0279
credit_history=no credits/all paid -0.8129
credit_history=all paid            -0.9562
credit_history=existing paid       -0.2268
credit_history=delayed previously   0.0403
```

credit_history=critical/other existing credit 0.6229
purpose=new car -0.692
purpose=used car 0.9744
purpose=furniture/equipment 0.0996
purpose=radio/tv 0.1996
purpose=domestic appliance -0.1692
purpose=repairs -0.4756
purpose=education -0.7283
purpose=vacation 0
purpose=retraining 1.3674
purpose=business 0.0481
purpose=other 0.7967
credit_amount -0.0001
savings_status=<100 -0.4402
savings_status=100<=X<500 -0.0825
savings_status=500<=X<1000 -0.0641
savings_status=>=1000 0.8989
savings_status=no known savings 0.5064
employment=unemployed -0.2934
employment=<1 -0.2265
employment=1<=X<4 -0.1106
employment=4<=X<7 0.5376
employment=>=7 -0.0168
installment_commitment -0.3301
personal_status=male div/sep -0.4922
personal_status=female div/dep/mar -0.2168
personal_status=male single 0.3238
personal_status=male mar/wid -0.1252
personal_status=female single 0
other_parties=none -0.1798
other_parties=co applicant -0.6158
other_parties=guarantor 0.7988
residence_since -0.0048
property_magnitude=real estate 0.2572
property_magnitude=life insurance -0.0242
property_magnitude=car 0.0627
property_magnitude=no known property -0.4732
age 0.0145
other_payment_plans=bank -0.3273
other_payment_plans=stores -0.2041
other_payment_plans=none 0.3191
housing=rent -0.3497
housing=own 0.0939
housing=for free 0.3341
existing_credits -0.2721
job=unemp/unskilled non res 0.5096
job=unskilled resident -0.0265
job=skilled -0.0451

job=high qualif/self emp/mgmt	0.0301
num_dependents	-0.2647
own_telephone=yes	0.3
foreign_worker=no	1.3922
Intercept	3.1983

Odds Ratios...

Variable	Class good
checking_status=<0	0.4593
checking_status=0<=X<200	0.6682
checking_status=>=200	1.2064
checking_status=no checking	2.5443
duration	0.9725
credit_history=no credits/all paid	0.4436
credit_history=all paid	0.3843
credit_history=existing paid	0.7971
credit_history=delayed previously	1.0411
credit_history=critical/other existing credit	1.8643
purpose=new car	0.5006
purpose=used car	2.6496
purpose=furniture/equipment	1.1047
purpose=radio/tv	1.2209
purpose=domestic appliance	0.8443
purpose=repairs	0.6215
purpose=education	0.4827
purpose=vacation	1
purpose=retraining	3.925
purpose=business	1.0492
purpose=other	2.2182
credit_amount	0.9999
savings_status=<100	0.6439
savings_status=100<=X<500	0.9208
savings_status=500<=X<1000	0.9379
savings_status=>=1000	2.457
savings_status=no known savings	1.6593
employment=unemployed	0.7457
employment=<1	0.7973
employment=1<=X<4	0.8953
employment=4<=X<7	1.7119
employment=>=7	0.9834
installment_commitment	0.7189
personal_status=male div/sep	0.6113
personal_status=female div/dep/mar	0.8051
personal_status=male single	1.3824

personal_status=male mar/wid 0.8824
personal_status=female single 1
other_parties=none 0.8354
other_parties=co applicant 0.5402
other_parties=guarantor 2.2229
residence_since 0.9952
property_magnitude=real estate 1.2933
property_magnitude=life insurance 0.9761
property_magnitude=car 1.0647
property_magnitude=no known property 0.623
age 1.0146
other_payment_plans=bank 0.7209
other_payment_plans=stores 0.8154
other_payment_plans=none 1.3758
housing=rent 0.7049
housing=own 1.0984
housing=for free 1.3967
existing_credits 0.7618
job=unemp/unskilled non res 1.6647
job=unskilled resident 0.9738
job=skilled 0.9559
job=high qualif/self emp/mgmt 1.0306
num_dependents 0.7674
own_telephone=yes 1.3499
foreign_worker=no 4.0237

Time taken to build model: 0.22 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 752 75.2 %
Incorrectly Classified Instances 248 24.8 %
Kappa statistic 0.375
Mean absolute error 0.3098
Root mean squared error 0.4087
Relative absolute error 73.727 %
Root relative squared error 89.1751 %
Total Number of Instances 1000
=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.864	0.510	0.798	0.864	0.830	0.379	0.785	0.883	good
0.490	0.136	0.607	0.490	0.542	0.379	0.785	0.599	bad
Weighted Avg. 0.752	0.398	0.741	0.752	0.744	0.379	0.785	0.798	

=== Confusion Matrix ===

a	b	<-- classified as
605	95	a = good
153	147	b = bad

```
  a   b   <-- classified as
605  95 |   a = good
153 147 |   b = bad
```

El modelo utiliza **Regresión Logística**, es un modelo probabilístico basado en una combinación lineal ponderada de los atributos de entrada. Los pasos para el proceso de decisión son:

- **Pesos en los atributos:** Cada atributo tiene un peso asignado, pueden ser negativo o positivos
- **Cálculo de probabilidades:** El modelo combina los valores de los atributos con los coeficientes para calcular una puntuación lineal. Luego aplica la función sigmoide para convertir esa puntuación en una probabilidad entre 0 y 1.
- **Clasificación:** Si la probabilidad de "good" supera un umbral, clasifica la instancia como "good"; de lo contrario, como "bad".

Atributos importantes

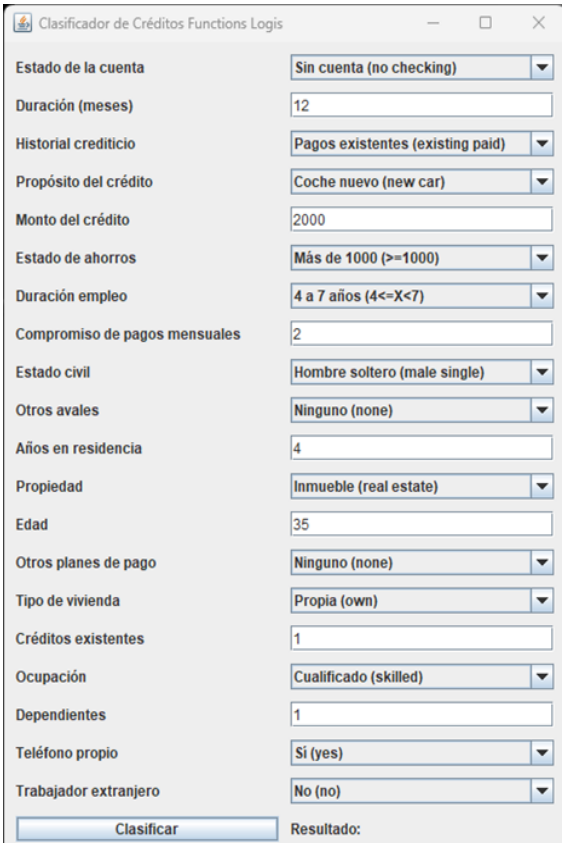
- **Para la clase “good” (coeficientes positivos)**
- checking_status=no checking (Coef: 0.9339): Las personas sin una cuenta de cheques tienen más probabilidades de ser "good".
- foreign_worker=no (Coef: 1.3922): Trabajadores no extranjeros están más asociados con "good".
- purpose=retraining (Coef: 1.3674): Personas que solicitan crédito para "retraining" tienen mayores probabilidades de ser "good".
- **Para la clase “bad” (coeficientes negativos)**
- credit_history=all paid (Coef: -0.9562): Personas con historial de crédito "todo pagado" se asocian más con "bad".
- duration (Coef: -0.0279): Créditos con mayor duración tienden a clasificarse como "bad".
- installment_commitment (Coef: -0.3301): Mayores compromisos en las cuotas de pago se relacionan con "bad".

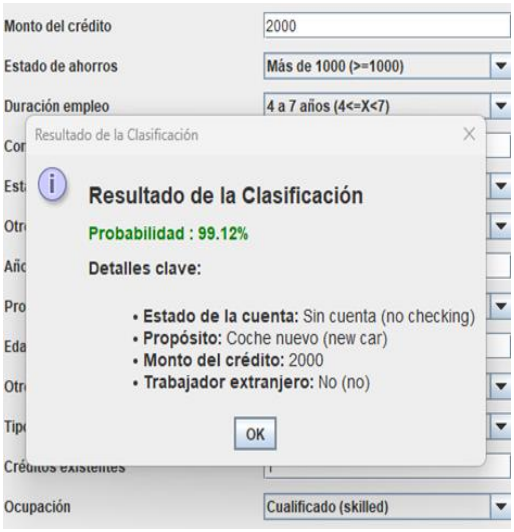
Precisión del modelo

- **Para la clase “good”**
- Tasa de acierto (TP Rate): 86.4%. Clasifica correctamente a la mayoría de los "good".
- Tasa de acierto (TP Rate): 86.4%. Clasifica correctamente a la mayoría de los "good".
- **Para la clase “bad”**
- Tasa de acierto (TP Rate): 49%. Tiene dificultades para identificar a los "bad".
- Precisión: 60.7%. Las predicciones de "bad" son menos confiables.

Conclusión: El modelo es confiable para predecir casos de "good", pero tiene menor precisión para los casos "bad".

Datos de prueba para el análisis de resultados
Ejemplo: PARA LA CLASE "GOOD"



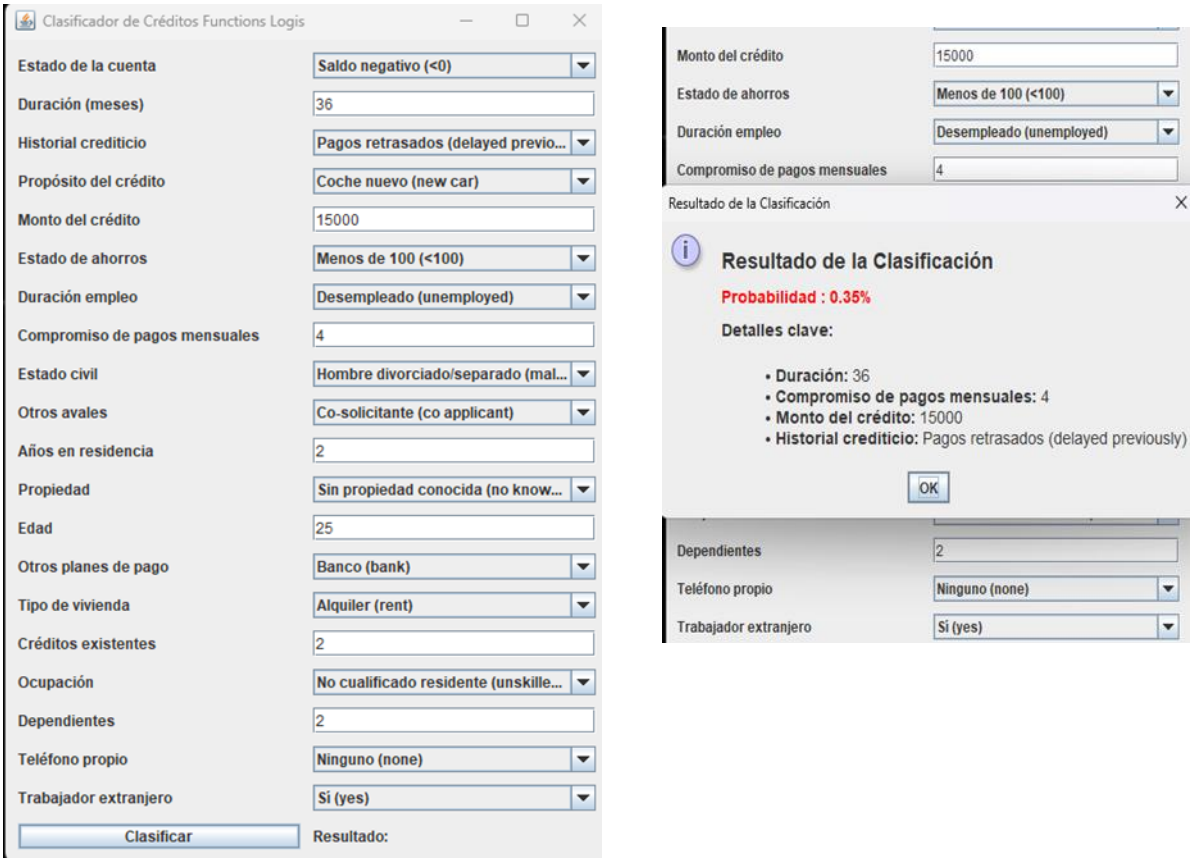


En este caso el resultado obtenido es Good (de color verde) los que indica que el perfil es adecuado según el modelo.

El cliente fue clasificado como good debido a la combinación favorable de los siguientes atributos más influyentes:

- checking_status=no checking: Al no tener una cuenta de cheques, el cliente está más asociado con un perfil good, ya que este grupo estadísticamente presenta mejores resultados crediticios.
- foreign_worker=no: El hecho de no ser un trabajador extranjero refuerza la clasificación positiva, indicando mayor estabilidad percibida en el contexto local.
- purpose=retraining: Solicitar crédito para retraining se asocia con un propósito que estadísticamente muestra menores niveles de riesgo.

Ejemplo: PARA LA CLASE “BAD”



En este caso el resultado obtenido es Bad (de color rojo) los que indica que el perfil es no es adecuado según el modelo.

El resultado bad refleja un perfil crediticio con mayor riesgo, influenciado por los siguientes atributos:

- **credit_history=all paid:** Aunque podría parecer contraintuitivo, un historial de crédito marcado como "todo pagado" puede estar asociado con un perfil de bajo dinamismo financiero, lo cual inclina la clasificación hacia bad.
- **duration:** Un crédito con una duración prolongada incrementa las probabilidades de ser clasificado como bad, ya que representa un compromiso financiero más extenso y riesgoso.
- **installment_commitment:** El cliente tiene un compromiso alto con las cuotas de pago, lo que refuerza la percepción de dificultad para manejar cargas adicionales.

Meta Randomizable Filtered Classifier

=== Run information ===

Scheme: weka.classifiers.meta.RandomizableFilteredClassifier -F "weka.filters.unsupervised.attribute.RandomProjection -N 10 -R 42 -D Sparse1" -S 1 -W weka.classifiers.lazy.IBk -- -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Relation: german_credit

Instances: 1000

Attributes: 21

checking_status
duration
credit_history
purpose
credit_amount
savings_status
employment
installment_commitment
personal_status
other_parties
residence_since
property_magnitude
age
other_payment_plans
housing
existing_credits
job
num_dependents
own_telephone
foreign_worker
class

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

RandomizableFilteredClassifier using weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\"" on data filtered through weka.filters.unsupervised.attribute.RandomProjection -N 10 -R -1085589929 -D Sparse1

Filtered Header

@relation german_credit-weka.filters.supervised.attribute.NominalToBinary-weka.filters.unsupervised.attribute.RandomProjection-N10-R-1085589929-DSparse1

@attribute K1 numeric
@attribute K2 numeric
@attribute K3 numeric
@attribute K4 numeric
@attribute K5 numeric
@attribute K6 numeric
@attribute K7 numeric
@attribute K8 numeric
@attribute K9 numeric

@attribute K10 numeric
@attribute class {good,bad}

@data

Classifier Model
IB1 instance-based classifier
using 1 nearest neighbour(s) for classification

Time taken to build model: 0.06 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	656	65.6 %
Incorrectly Classified Instances	344	34.4 %
Kappa statistic	0.1731	
Mean absolute error	0.3443	
Root mean squared error	0.5859	
Relative absolute error	81.9525 %	
Root relative squared error	127.8463 %	
Total Number of Instances	1000	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.761	0.590	0.751	0.761	0.756	0.173	0.586	0.739	good
	0.410	0.239	0.424	0.410	0.417	0.173	0.586	0.351	bad
Weighted Avg.	0.656	0.485	0.653	0.656	0.654	0.173	0.586	0.622	

=== Confusion Matrix ===

a	b	<-- classified as
533	167	a = good
177	123	b = bad

El modelo utilizado es un **RandomizableFilteredClassifier**, que combina un filtro de proyección aleatoria con el algoritmo **IBk** (k-vecinos más cercanos) para tomar decisiones basadas en la instancia más cercana (ya que $k=1k = 1$k=1). Los pasos para la toma de decisión son:

- **Proyección Aleatoria (RandomProjection):** Antes de aplicar el clasificador, los atributos originales se transforman mediante un filtro de proyección aleatoria que reduce la dimensionalidad del conjunto de datos a 10 atributos proyectados (K1 a K10).
- **Clasificación por Vecinos (IBk):** Despues de la proyección, el clasificador busca la instancia más cercana (k=1) al punto que desea clasificar en el espacio transformado. La cercanía se mide utilizando la distancia euclidiana considerando los 10 atributos proyectados (K1 a K10).
- **Decisión basada en la clase de vecino mas cercado:** La clase del vecino más cercano (ya sea good o bad) se asigna a la instancia analizada.

Atributos tomados en cuenta

Debido a la proyección aleatoria, los atributos originales (como checking_status, duration, credit_history, purpose, credit_amount, savings_status, employment, etc.) son combinados de manera lineal y transformados en los nuevos atributos (K1, K2, ..., K10).

- Cada uno de los atributos originales influye indirectamente en la decisión, pero no es posible identificar un impacto individual directo después de la transformación

Precisión del Modelo

- **Precisión general:** El modelo clasifica correctamente el **65.6%** de las instancias. Aunque esto es significativamente mejor que el azar, no es ideal para aplicaciones críticas
- **Para la clase "good":**

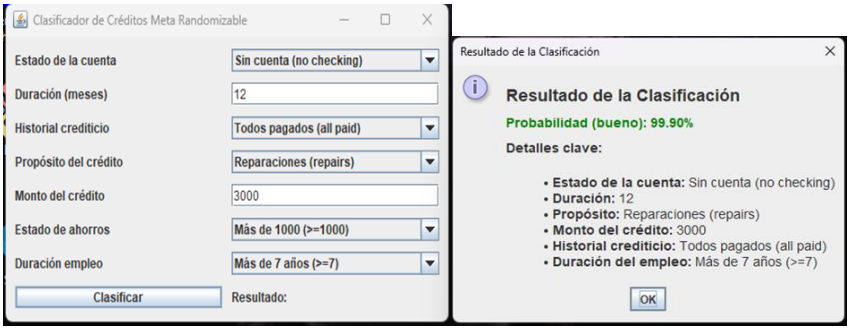
- Tasa de acierto (TP Rate): 76.1%
- Precisión: 75.1%
- Esto significa que el modelo identifica correctamente a la mayoría de las instancias good.

- **Para la clase "bad"** :

- Tasa de acierto (TP Rate): 41.0%
- Precisión: 42.4%
- Tiene más dificultades para identificar correctamente las instancias bad, posiblemente debido al desbalance entre las clases o una proyección que no captura bien la estructura de los datos.

Conclusión: El modelo clasifica usando el vecino más cercano, con un 65.6% de precisión. Es más efectivo en la clase good (76.1%) que en bad (41%), pero su baja fiabilidad (Kappa: 0.173) sugiere margen de mejora.

Datos de prueba para el análisis de resultados
Ejemplo: PARA LA CLASE “GOOD”

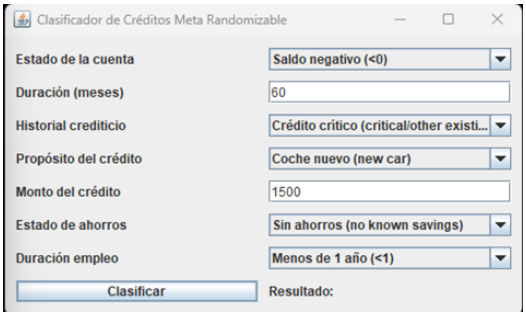


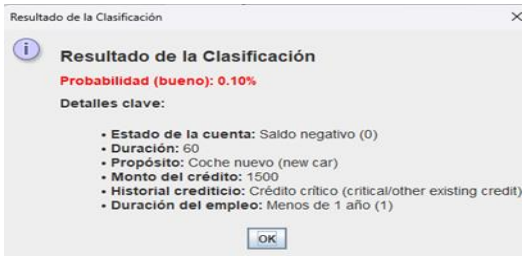
En este caso, el resultado obtenido es Good (de color verde), lo que indica que el perfil es adecuado según el modelo.

El cliente fue clasificado como Good debido a la combinación favorable de los siguientes atributos más influyentes:

- Estado de la cuenta: "no checking", asociado con buenos hábitos financieros.
- Propósito del crédito: "retraining", considerado una razón sólida y de bajo riesgo.
- Historial crediticio: "all paid", que refuerza la confiabilidad del solicitante.
- Estado de ahorros: Ahorros superiores a 1000, lo que demuestra estabilidad financiera.

Ejemplo: PARA LA CLASE “BAD”



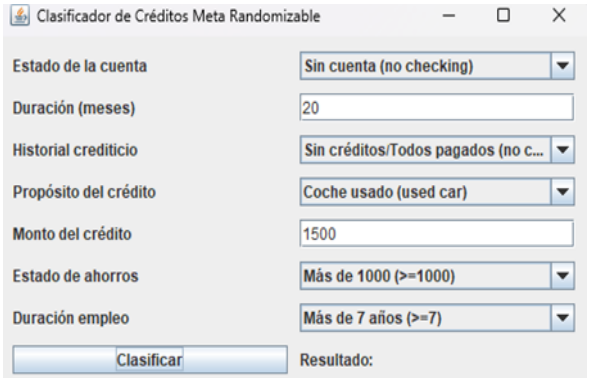


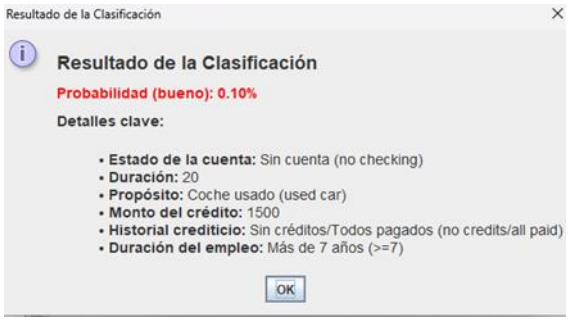
En este caso, el resultado obtenido es Bad (de color rojo), lo que indica que el perfil presenta un mayor riesgo según el modelo.

El cliente fue clasificado como Bad debido a los siguientes atributos más influyentes:

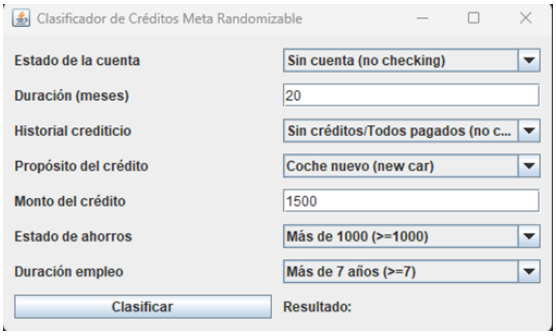
- Estado de la cuenta: "<0", lo que refleja posibles problemas financieros previos.
- Duración del crédito: 60 meses, asociado con un compromiso a largo plazo y mayor riesgo.
- Historial crediticio: "critical/other existing credit", que indica antecedentes crediticios negativos.
- Estado de ahorros: "no known savings", lo que implica falta de respaldo económico.

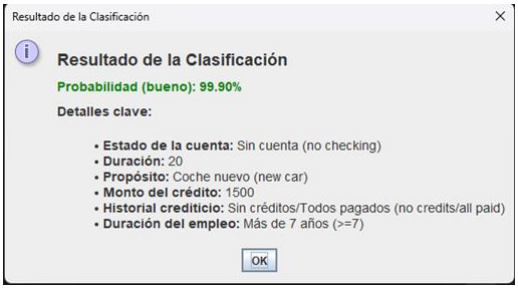
Otro ejemplo:





En este caso, el resultado obtenido es Bad (de color rojo), lo que indica que el perfil presenta un mayor riesgo según el modelo. Solo por el propósito del crédito en este caso es **Coche usado(used car),** me da un resultado **"bad".** Pero si el valor de ese atributo los cambiamos a **Coche nuevo (new car)** me da un resultado **"good"**





BAYESNET.

```
=== Run information ===

Scheme:       weka.classifiers.bayes.BayesNet -D -Q
weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E
weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5
Relation:     german_credit
Instances:    1000
Attributes:   21
              checking_status
              duration
              credit_history
              purpose
              credit_amount
              savings_status
              employment
              installment_commitment
              personal_status
              other_parties
              residence_since
              property_magnitude
              age
              other_payment_plans
              housing
              existing_credits
              job
              num_dependents
              own_telephone
              foreign_worker
              class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Bayes Network Classifier
not using ADTree
#attributes=21 #classindex=20
Network structure (nodes followed by parents)
checking_status(4): class
duration(2): class
credit_history(5): class
purpose(11): class
credit_amount(2): class
savings_status(5): class
employment(5): class
installment_commitment(1): class
```

personal_status(5): class
other_parties(3): class
residence_since(1): class
property_magnitude(4): class
age(1): class
other_payment_plans(3): class
housing(3): class
existing_credits(1): class
job(4): class
num_dependents(1): class
own_telephone(2): class
foreign_worker(2): class
class(2):
LogScore Bayes: -14834.516987143774
LogScore BDeu: -15015.677051924575
LogScore MDL: -15018.340534010807
LogScore ENTROPY: -14704.037668817118
LogScore AIC: -14795.037668817118

Time taken to build model: 0.07 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	755	75.5 %
Incorrectly Classified Instances	245	24.5 %
Kappa statistic	0.3893	
Mean absolute error	0.3101	
Root mean squared error	0.4187	
Relative absolute error	73.8033 %	
Root relative squared error	91.3646 %	
Total Number of Instances	1000	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,859	0,487	0,805	0,859	0,831	0,392	0,780	0,888	good
	0,513	0,141	0,609	0,513	0,557	0,392	0,780	0,557	bad
Weighted Avg.	0,755	0,383	0,746	0,755	0,749	0,392	0,780	0,789	

=== Confusion Matrix ===

a	b	<-- classified as
601	99	a = good
146	154	b = bad

El clasificador BayesNet es una implementación de Redes Bayesianas, que representa gráficamente relaciones probabilísticas entre variables. En este caso, esta siendo aplicado a un conjunto de datos llamados **german_credit,** relacionado con información crediticia de 1000 individuos.

Las Variables (atributos) se representan como nodos. Las relaciones de dependencias entre las variables se muestran como aristas dirigidas entre nodos. Cada nodo tiene una tabla de probabilidad condicional (CPT) que define las probabilidades de los valores del nodo en funciona de sus padres.

Componentes del Modelo:

- **Estimador Simple (SimpleEstimator):** Asigna probabilidades iniciales a los nodos basándose en un valor de suavizado (-A 0.5), lo que evita probabilidades nulas.
- **Búsqueda estructural (K2)**: Busca la estructura de la red explorando combinaciones de dependencias entre variables y seleccionando la que mejor se ajuste.

Clasificación:

- En este caso, clasifica a los individuos en dos categorías: "good" (buen crédito) y "bad" (mal crédito).
- Se utiliza para predecir el comportamiento crediticio basándose en atributos como edad, empleo, historial de crédito, etc.

La red muestra que variables son mas relevantes para predecir la clase y como están relacionadas entre sí. Puede ser usado por instituciones financieras para decidir si otorgar un préstamo o no.

Resultados Obtenidos.

- **Estructura de la red:** Se muestra que todas las variables dependen directamente de la clase (nodo class), lo que indica que la clase influye en cada atributo.

- **Desempeño del Modelo:**
  - **Precisión (Accuracy):** 75.5% (clasifica correctamente 755 de 1000 instancias).
  - **Kappa:** 0.3893 (moderada concordancia entre predicciones y realidad).
  - **ROC Area:** 0.780 para ambas clases, lo que indica una buena capacidad de diferenciación.

- **Errores:**
  - **Error absoluto medio (MAE):** 0.3101 (indica qué tan lejos están, en promedio, las predicciones de los valores reales).
  - **Error cuadrático medio (RMSE):** 0.4187 (pondera los errores más grandes, dándoles más importancia).

- **Confusión entre clases**:
  Clase “Good” (buen crédito):
    - **TPR (Tasa de verdaderos positivos):** 85.9% (correctamente clasificados como buenos).
    - **FPR (Tasa de falsos positivos):** 48.7% (erróneamente clasificados como malos).

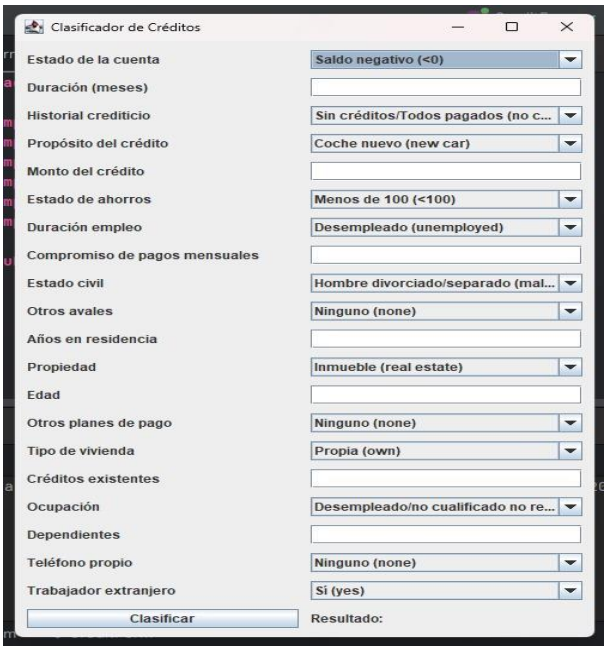
  Clase “bad” (Mal credito):
    - **TPR:** 51.3% (correctamente clasificados como malos).
    - **FPR:** 14.1% (erróneamente clasificados como buenos).

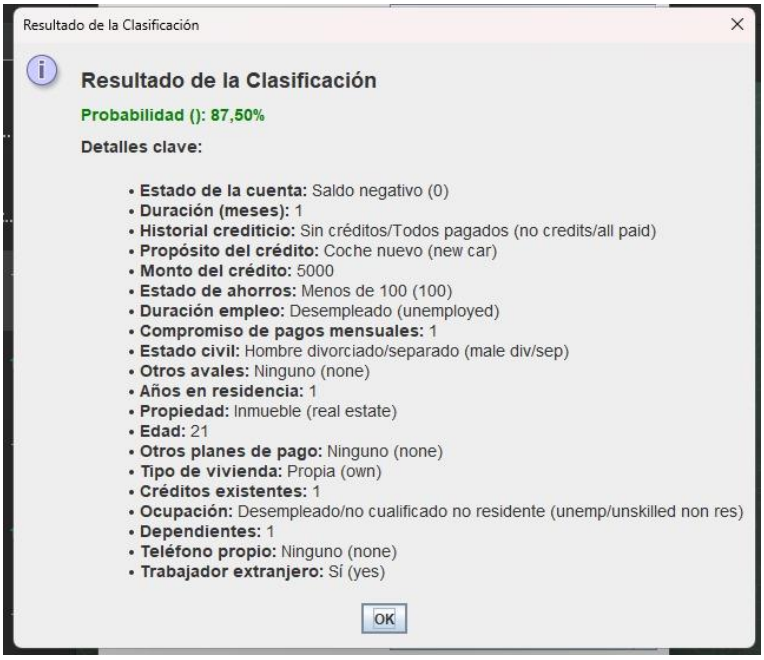
¿Es confiable?

- **Fortalezas.**
    - Es interpretativo, ya que la estructura de la red permite visualizar las relaciones entre variables.
    - Es robusto frente a datos incompletos, ya que utiliza probabilidades condicionales para inferir información faltante.
    - Maneja bien datos categóricos y numéricos.

- **Limitaciones.**
    - La precisión del 75.5% indica que hay margen de mejora, especialmente en la clase "bad", donde solo el 51.3% de los casos son clasificados correctamente.
    - El valor de Kappa (0.3893) sugiere una concordancia moderada, lo que indica que puede haber incertidumbre en las predicciones.
    - La confiabilidad depende mucho de la calidad y representatividad del conjunto de datos.

Este modelo sirve para clasificar clientes de acuerdo con su comportamiento crediticio usando probabilidades condicionales basadas en la red. Es razonablemente confiable para la clase "good", pero menos para "bad". Aunque puede ser útil para tareas iniciales de evaluación crediticia, su implementación en un sistema real debería complementarse con otras técnicas o análisis para mejorar su precisión.


Clasificador de Créditos
Estado de la cuenta
Saldo negativo (<0)
Duración (meses)
Historial crediticio
Sin créditos/Todos pagados (no c...
Propósito del crédito
Coche nuevo (new car)
Monto del crédito
Estado de ahorros
Menos de 100 (<100)
Duración empleo
Desempleado (unemployed)
Compromiso de pagos mensuales
Estado civil
Hombre divorciado/separado (mal...
Otros avales
Ninguno (none)
Años en residencia
Propiedad
Inmueble (real estate)
Edad
Otros planes de pago
Ninguno (none)
Tipo de vivienda
Propia (own)
Créditos existentes
Ocupación
Desempleado/no cualificado no re...
Dependientes
Teléfono propio
Ninguno (none)
Trabajador extranjero
Si (yes)
Clasificar
Resultado:


Resultado de la Clasificación
Resultado de la Clasificación
Probabilidad (): 87,50%
Detalles clave:
• Estado de la cuenta: Saldo negativo (0)
• Duración (meses): 1
• Historial crediticio: Sin créditos/Todos pagados (no credits/all paid)
• Propósito del crédito: Coche nuevo (new car)
• Monto del crédito: 5000
• Estado de ahorros: Menos de 100 (100)
• Duración empleo: Desempleado (unemployed)
• Compromiso de pagos mensuales: 1
• Estado civil: Hombre divorciado/separado (male div/sep)
• Otros avales: Ninguno (none)
• Años en residencia: 1
• Propiedad: Inmueble (real estate)
• Edad: 21
• Otros planes de pago: Ninguno (none)
• Tipo de vivienda: Propia (own)
• Créditos existentes: 1
• Ocupación: Desempleado/no cualificado no residente (unemp/unskilled non res)
• Dependientes: 1
• Teléfono propio: Ninguno (none)
• Trabajador extranjero: Sí (yes)
OK

J48.

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     german_credit
Instances:    1000
Attributes:   21
              checking_status
              duration
              credit_history
              purpose
              credit_amount
              savings_status
              employment
              installment_commitment
              personal_status
              other_parties
              residence_since
              property_magnitude
              age
              other_payment_plans
              housing
              existing_credits
              job
              num_dependents
              own_telephone
              foreign_worker
              class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

checking_status = <0
|   foreign_worker = yes
|   |   duration <= 11
|   |   |   existing_credits <= 1
|   |   |   |   property_magnitude = real estate: good (8.0/1.0)
|   |   |   |   property_magnitude = life insurance
|   |   |   |   |   own_telephone = none: bad (2.0)
|   |   |   |   |   own_telephone = yes: good (4.0)
|   |   |   |   property_magnitude = car: good (2.0/1.0)
|   |   |   |   property_magnitude = no known property: bad (3.0)
|   |   |   existing_credits > 1: good (14.0)
```

```
|   |   duration > 11
|   |   |   job = unemp/unskilled non res: bad (5.0/1.0)
|   |   |   job = unskilled resident
|   |   |   |   purpose = new car
|   |   |   |   |   own_telephone = none: bad (10.0/2.0)
|   |   |   |   |   own_telephone = yes: good (2.0)
|   |   |   |   purpose = used car: bad (1.0)
|   |   |   |   purpose = furniture/equipment
|   |   |   |   |   employment = unemployed: good (0.0)
|   |   |   |   |   employment = <1: bad (3.0)
|   |   |   |   |   employment = 1<=X<4: good (4.0)
|   |   |   |   |   employment = 4<=X<7: good (1.0)
|   |   |   |   |   employment = >=7: good (2.0)
|   |   |   |   purpose = radio/tv
|   |   |   |   |   existing_credits <= 1: bad (10.0/3.0)
|   |   |   |   |   existing_credits > 1: good (2.0)
|   |   |   |   purpose = domestic appliance: bad (1.0)
|   |   |   |   purpose = repairs: bad (1.0)
|   |   |   |   purpose = education: bad (1.0)
|   |   |   |   purpose = vacation: bad (0.0)
|   |   |   |   purpose = retraining: good (1.0)
|   |   |   |   purpose = business: good (3.0)
|   |   |   |   purpose = other: good (1.0)
|   |   |   job = skilled
|   |   |   |   other_parties = none
|   |   |   |   |   duration <= 30
|   |   |   |   |   |   savings_status = <100
|   |   |   |   |   |   |   credit_history = no credits/all paid: bad (8.0/1.0)
|   |   |   |   |   |   |   credit_history = all paid: bad (6.0)
|   |   |   |   |   |   |   credit_history = existing paid
|   |   |   |   |   |   |   |   own_telephone = none
|   |   |   |   |   |   |   |   |   existing_credits <= 1
|   |   |   |   |   |   |   |   |   |   property_magnitude = real estate
|   |   |   |   |   |   |   |   |   |   |   age <= 26: bad (5.0)
|   |   |   |   |   |   |   |   |   |   |   age > 26: good (2.0)
|   |   |   |   |   |   |   |   |   |   property_magnitude = life insurance: bad (7.0/2.0)
|   |   |   |   |   |   |   |   |   |   property_magnitude = car
|   |   |   |   |   |   |   |   |   |   |   credit_amount <= 1386: bad (3.0)
|   |   |   |   |   |   |   |   |   |   |   credit_amount > 1386: good (11.0/1.0)
|   |   |   |   |   |   |   |   |   |   property_magnitude = no known property: good (2.0)
|   |   |   |   |   |   |   |   |   existing_credits > 1: bad (3.0)
|   |   |   |   |   |   |   |   own_telephone = yes: bad (5.0)
|   |   |   |   |   |   |   credit_history = delayed previously: bad (4.0)
|   |   |   |   |   |   |   credit_history = critical/other existing credit: good (14.0/4.0)
|   |   |   |   |   |   savings_status = 100<=X<500
|   |   |   |   |   |   |   credit_history = no credits/all paid: good (0.0)
|   |   |   |   |   |   |   credit_history = all paid: good (1.0)
```

```
|  |  |  |  |  |  |  credit_history = existing paid: bad (3.0)
|  |  |  |  |  |  |  credit_history = delayed previously: good (0.0)
|  |  |  |  |  |  |  credit_history = critical/other existing credit: good (2.0)
|  |  |  |  |  |  savings_status = 500<=X<1000: good (4.0/1.0)
|  |  |  |  |  |  savings_status = >=1000: good (4.0)
|  |  |  |  |  |  savings_status = no known savings
|  |  |  |  |  |  |  existing_credits <= 1
|  |  |  |  |  |  |  |  own_telephone = none: bad (9.0/1.0)
|  |  |  |  |  |  |  |  own_telephone = yes: good (4.0/1.0)
|  |  |  |  |  |  |  existing_credits > 1: good (2.0)
|  |  |  |  |  duration > 30: bad (30.0/3.0)
|  |  |  |  other_parties = co applicant: bad (7.0/1.0)
|  |  |  |  other_parties = guarantor: good (12.0/3.0)
|  |  |  job = high qualif/self emp/mgmt: good (30.0/8.0)
|  foreign_worker = no: good (15.0/2.0)
checking_status = 0<=X<200
|  credit_amount <= 9857
|  |  savings_status = <100
|  |  |  other_parties = none
|  |  |  |  duration <= 42
|  |  |  |  |  personal_status = male div/sep: bad (8.0/2.0)
|  |  |  |  |  personal_status = female div/dep/mar
|  |  |  |  |  |  purpose = new car: bad (5.0/1.0)
|  |  |  |  |  |  purpose = used car: bad (1.0)
|  |  |  |  |  |  purpose = furniture/equipment
|  |  |  |  |  |  |  duration <= 10: bad (3.0)
|  |  |  |  |  |  |  duration > 10
|  |  |  |  |  |  |  |  duration <= 21: good (6.0/1.0)
|  |  |  |  |  |  |  |  duration > 21: bad (2.0)
|  |  |  |  |  |  purpose = radio/tv: good (8.0/2.0)
|  |  |  |  |  |  purpose = domestic appliance: good (0.0)
|  |  |  |  |  |  purpose = repairs: good (1.0)
|  |  |  |  |  |  purpose = education: good (4.0/2.0)
|  |  |  |  |  |  purpose = vacation: good (0.0)
|  |  |  |  |  |  purpose = retraining: good (0.0)
|  |  |  |  |  |  purpose = business
|  |  |  |  |  |  |  residence_since <= 2: good (3.0)
|  |  |  |  |  |  |  residence_since > 2: bad (2.0)
|  |  |  |  |  |  purpose = other: good (0.0)
|  |  |  |  |  personal_status = male single: good (52.0/15.0)
|  |  |  |  |  personal_status = male mar/wid
|  |  |  |  |  |  duration <= 10: good (6.0)
|  |  |  |  |  |  duration > 10: bad (10.0/3.0)
|  |  |  |  |  personal_status = female single: good (0.0)
|  |  |  |  duration > 42: bad (7.0)
|  |  |  other_parties = co applicant: good (2.0)
|  |  |  other_parties = guarantor
```

```
|  |  |  |  purpose = new car: bad (2.0)
|  |  |  |  purpose = used car: good (0.0)
|  |  |  |  purpose = furniture/equipment: good (0.0)
|  |  |  |  purpose = radio/tv: good (18.0/1.0)
|  |  |  |  purpose = domestic appliance: good (0.0)
|  |  |  |  purpose = repairs: good (0.0)
|  |  |  |  purpose = education: good (0.0)
|  |  |  |  purpose = vacation: good (0.0)
|  |  |  |  purpose = retraining: good (0.0)
|  |  |  |  purpose = business: good (0.0)
|  |  |  |  purpose = other: good (0.0)
|  |  savings_status = 100<=X<500
|  |  |  purpose = new car: bad (15.0/5.0)
|  |  |  purpose = used car: good (3.0)
|  |  |  purpose = furniture/equipment: bad (4.0/1.0)
|  |  |  purpose = radio/tv: bad (8.0/2.0)
|  |  |  purpose = domestic appliance: good (0.0)
|  |  |  purpose = repairs: good (2.0)
|  |  |  purpose = education: good (0.0)
|  |  |  purpose = vacation: good (0.0)
|  |  |  purpose = retraining: good (0.0)
|  |  |  purpose = business
|  |  |  |  housing = rent
|  |  |  |  |  existing_credits <= 1: good (2.0)
|  |  |  |  |  existing_credits > 1: bad (2.0)
|  |  |  |  housing = own: good (6.0)
|  |  |  |  housing = for free: bad (1.0)
|  |  |  purpose = other: good (1.0)
|  |  savings_status = 500<=X<1000: good (11.0/3.0)
|  |  savings_status = >=1000: good (13.0/3.0)
|  |  savings_status = no known savings: good (41.0/5.0)
|  credit_amount > 9857: bad (20.0/3.0)
checking_status = >=200: good (63.0/14.0)
checking_status = no checking: good (394.0/46.0)

Number of Leaves  :         103

Size of the tree :      140


Time taken to build model: 0.1 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         705               70.5    %
Incorrectly Classified Instances       295               29.5    %
```

```
Kappa statistic                  0.2467
Mean absolute error              0.3467
Root mean squared error          0.4796
Relative absolute error         82.5233 %
Root relative squared error     104.6565 %
Total Number of Instances       1000

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,840    0,610    0,763      0,840   0,799      0,251    0,639     0,746     good
               0,390    0,160    0,511      0,390   0,442      0,251    0,639     0,449     bad
Weighted Avg.  0,705    0,475    0,687      0,705   0,692      0,251    0,639     0,657

=== Confusion Matrix ===

   a   b   <-- classified as
 588 112 |   a = good
 183 117 |   b = bad
```

El algoritmo **J48** es una implementación del árbol de decisión **C4.5** en Weka, usado para tareas de clasificación. Construye un modelo de árbol de decisión basado en atributos del conjunto de datos para predecir a qué clase pertenece cada instancia. Este modelo en particular fue aplicado al conjunto de datos german_credit para clasificar a individuos como buenos (good) o malos (bad) clientes crediticios.

¿Cómo funciona?

- **Es un árbol de decisión:**
  - Divide los datos en subconjuntos más pequeños usando reglas basadas en los valores de los atributos.
  - Cada división intenta maximizar la pureza de las clases en los subconjuntos.

- **Proceso de construcción:**
  - Comienza con todos los datos en la raíz del árbol.
  - Selecciona el atributo que mejor divide los datos basándose en una métrica (e.g., ganancia de información o índice de Gini).
  - Divide los datos y repite el proceso en cada nodo hijo.
  - **Se detiene cuando:**
    Los nodos contienen menos instancias que un umbral mínimo (-M 2, mínimo de 2 instancias en este caso).
    No se puede mejorar la pureza al dividir.

- **Poda del árbol:**
    - Tras construir el árbol completo, elimina ramas innecesarias que no contribuyen significativamente a la precisión (-C 0.25, con un nivel de confianza del 25%).

Clasificación.
- Identifica patrones en los datos para clasificar a clientes como buenos o malos.
- Las reglas generadas son interpretables y pueden ser utilizadas para la toma de decisiones.

Interpretación de relaciones.
- Los caminos desde la raíz hasta las hojas representan reglas claras basadas en atributos.
- Ayuda a comprender qué factores (e.g., historial crediticio, cantidad de crédito) influyen más en la clasificación.

Aplicaciones prácticas.
- Decisiones crediticias en instituciones financieras.
- Evaluación de riesgos.
- Identificación de clientes problemáticos.

Resultados Obtenidos.
Estructura del árbol:
- **Tamaño del árbol:**
    - Número de hojas: 103.
    - Tamaño total del árbol: 140 nodos.
    - Esto indica un árbol relativamente grande, lo que sugiere reglas complejas en los datos.

- **Ramas principales:**
    - El atributo más importante es checking_status (estado de la cuenta corriente), que divide inicialmente los datos.
    - Otros Atributos relevantes que incluye:
      duration (duración del crédito).
      foreign_worker (si el cliente es trabajador extranjero).
      credit_amount (cantidad de crédito solicitado).
      property_magnitude (tipo de propiedad del cliente).
      employment (estatus laboral).

Desempeño del modelo.
- **Precisión general:**
    - **Clasifica correctamente el 70.5% de las instancias (705 de 1000).**
    - **Esto indica un desempeño aceptable pero no excelente.**

- **Kappa Statistic:**
    - 0.2467, lo que indica una baja concordancia entre las predicciones y las clases reales más allá del azar.

- **Errores:**
    - Error absoluto medio (MAE): 0.3467.
    - Error cuadrático medio (RMSE): 0.4796.
    - Estos valores indican que el modelo comete errores relativamente significativos en las predicciones.

Rendimiento de la clase.

Metrica	Clase “GOOD”	Clase “BAD”
TP Rate (Recall)	0.840	0.390
FP Rate	0.610	0.160
Precisión	0.763	0.511
F-Measure	0.799	0.442
ROC Area	0.639	0.639

- **Clase "good" (buen crédito):**
    - El modelo identifica correctamente el 84% de los buenos clientes.
    - Tiene un alto índice de falsos positivos (61%), clasificando clientes malos como buenos.

- **Clase "bad" (mal crédito):**
    - Solo identifica correctamente el 39% de los malos clientes.
    - Baja precisión para esta clase, lo que sugiere problemas con datos desbalanceados.

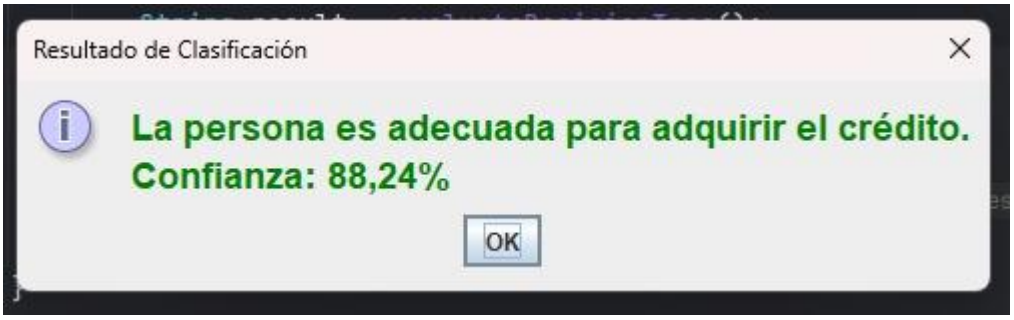
Fortalezas.

- Las reglas son fáciles de interpretar y explicar.
- Identifica correctamente la mayoría de los buenos clientes.
- El modelo se genera rápidamente (0.1 segundos).

Limitaciones.

- El desempeño para la clase "bad" es deficiente, con un TP Rate de solo 39%.
- Alto índice de falsos positivos en la clase "good" (61%).
- La métrica Kappa (0.2467) sugiere que el modelo tiene una confiabilidad limitada más allá del azar.

Este modelo J48 es útil para clasificar clientes crediticios, pero tiene limitaciones significativas para la clase "bad". Aunque su interpretabilidad lo hace ideal para contextos financieros, su precisión podría no ser suficiente para decisiones críticas sin ajustes adicionales. Es necesario complementarlo con técnicas avanzadas o mejorar los datos de entrada para aumentar su confiabilidad.


Resultado de Clasificación
La persona es adecuada para adquirir el crédito.
Confianza: 88,24%
OK

LAZY - IBK

=== Run information ===

Scheme: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation: german_credit
Instances: 1000
Attributes: 21
checking_status
duration
credit_history
purpose
credit_amount
savings_status
employment
installment_commitment
personal_status
other_parties
residence_since
property_magnitude
age
other_payment_plans
housing
existing_credits
job
num_dependents
own_telephone
foreign_worker
class
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	720	72	%
Incorrectly Classified Instances	280	28	%

Kappa statistic 0.3243
Mean absolute error 0.2805
Root mean squared error 0.5286
Relative absolute error 66.7546 %
Root relative squared error 115.3422 %
Total Number of Instances 1000

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,810	0,490	0,794	0,810	0,802	0,325	0,660	0,776	good
	0,510	0,190	0,535	0,510	0,522	0,325	0,660	0,420	bad
Weighted Avg.	0,720	0,400	0,716	0,720	0,718	0,325	0,660	0,669	

=== Confusion Matrix ===

a	b	<-- classified as
567	133	a = good
147	153	b = bad

Descripción del Algoritmo Lazy-IBK

El algoritmo **Lazy-IBK** (parte de la categoría de clasificadores "Lazy" en WEKA) es una implementación de **K-Nearest Neighbors (KNN)**, un modelo de aprendizaje supervisado utilizado para clasificación y regresión. Es "Lazy" porque:

- No construye un modelo explícito durante el entrenamiento.
- En su lugar, almacena las instancias de entrenamiento y realiza los cálculos necesarios para clasificar una nueva instancia en el momento de la consulta.

Parámetros del Algoritmo Lazy-IBK en tu proyecto

- **`-K 1`**: El valor de `K` indica el número de vecinos más cercanos que se utilizarán para clasificar una instancia. Aquí se usa **1 vecino**.
- **`-W 0`**: No hay ponderación en los vecinos (todas las instancias tienen el mismo peso).
- **`-A "weka.core.neighboursearch.LinearNNSearch`**: Indica que se usó búsqueda de vecinos lineal, lo que significa que todos los puntos del conjunto de datos son examinados para encontrar el vecino más cercano.
- **`weka.core.EuclideanDistance -R first-last`**: Utiliza la distancia euclidiana como métrica para medir la similitud entre instancias considerando todos los atributos (`first-last`).

Ventajas del modelo:

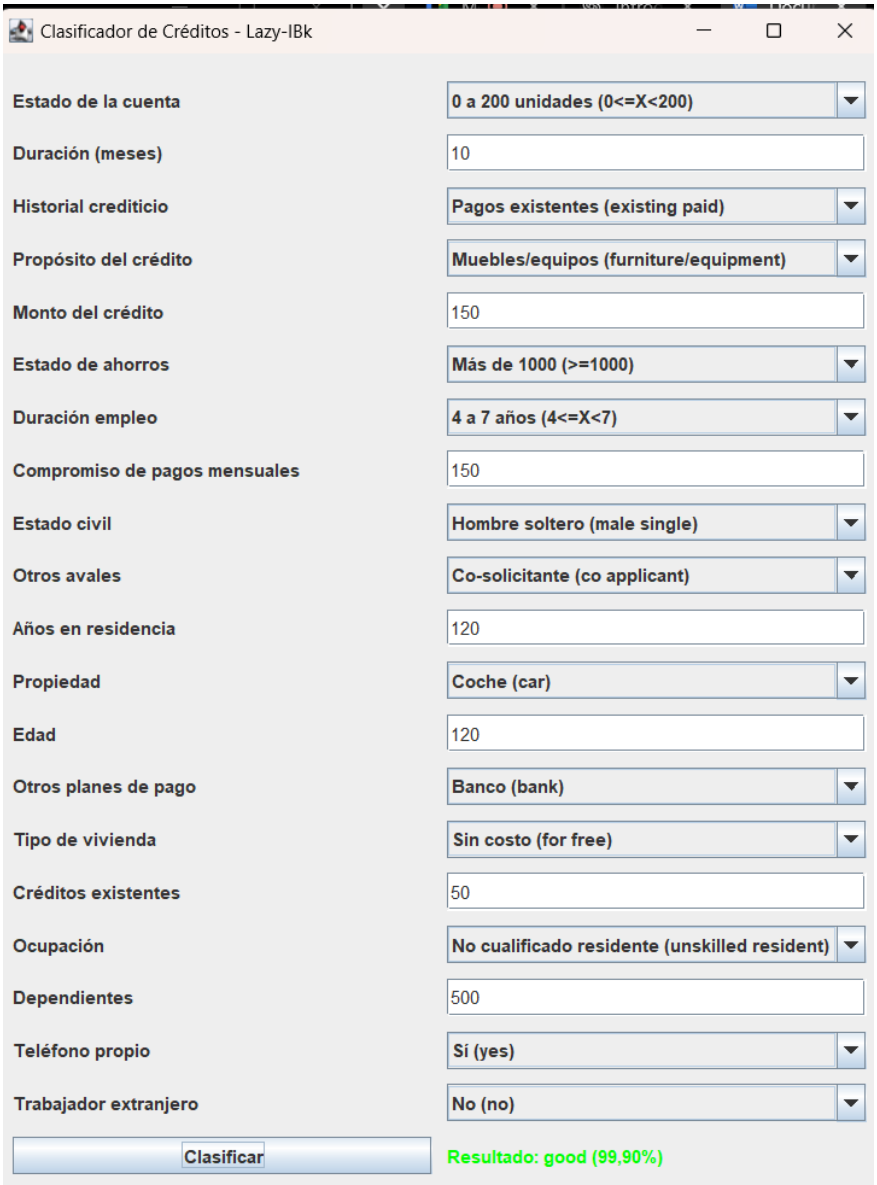
- Fácil de interpretar y entender.

- Muy efectivo en problemas donde las clases tienen fronteras bien definidas.

Limitaciones:

- El tiempo de clasificación puede ser lento con grandes datasets (debido a la búsqueda lineal).
- Sensible a datos ruidosos y desequilibrio de clases.

Resultados del Modelo



Resumen General

- **Número de instancias:** 1000.
- **Atributos:** 21 (incluyendo la clase `class`, con las categorías `good` y `bad`).
- **Validación cruzada:** Se utilizó una validación cruzada estratificada de 10 particiones (**10-fold cross-validation**) para evaluar el rendimiento del modelo.

Desempeño del Modelo

1. **Precisión Global:**
    a. El modelo clasificó correctamente el **72%** de las instancias (**720 correctas de 1000**).
    b. Esto implica que el **28%** de las instancias fueron clasificadas incorrectamente.
2. **Kappa Statistic (0.3243):**
    a. Esta métrica evalúa el acuerdo entre las predicciones del modelo y las clases reales, corrigiendo el azar.
    b. Un valor de **0.3243** indica un acuerdo moderado pero lejos de ser perfecto.
3. **Errores:**
    a. **Mean Absolute Error (MAE): 0.2805**, promedio del error absoluto en las predicciones.
    b. **Root Mean Squared Error (RMSE): 0.5286**, mide el error cuadrático medio.
4. **Curva ROC y Área Bajo la Curva (ROC Area):**
    a. El área bajo la curva ROC (AUC) es **0.660**. Esto indica que el modelo tiene un desempeño moderado para distinguir entre las clases `good` y `bad`.

Desempeño por Clase

El modelo tiene un mejor desempeño clasificando instancias de la clase `good` que `bad`, lo que refleja un posible **desequilibrio de clases** en el dataset o dificultad del modelo para identificar correctamente la clase `bad`.

Métrica	Good	Bad
TP Rate	0.810	0.510
FP Rate	0.490	0.190
Precision	0.794	0.535
Recall	0.810	0.510
F-Measure	0.802	0.522
ROC Area	0.660	0.660

- **TP Rate (True Positive Rate):**

La proporción de instancias correctamente clasificadas como su clase verdadera.

    - Clase `good`: 81% de las instancias fueron correctamente clasificadas.
    - Clase `bad`: Solo el 51% fueron correctamente clasificadas.
- **FP Rate (False Positive Rate):**

Proporción de instancias incorrectamente clasificadas como la otra clase.

    - Clase `good`: 49% de las instancias `bad` fueron clasificadas erróneamente como `good`.
- **Precision:**

Mide cuán precisas son las predicciones del modelo para cada clase.

    - Clase `good`: 79.4%.
    - Clase `bad`: 53.5%.
- **F-Measure:**

Es la media armónica entre la precisión y el recall.

    - Clase `good`: 80.2% (buen rendimiento).
    - Clase `bad`: 52.2% (rendimiento limitado).

Matriz de Confusión

Clasificado como	good	bad
good (real)	**567**	**133**
bad (real)	**147**	**153**

- **567 instancias** de la clase `good` fueron clasificadas correctamente, mientras que **133** fueron clasificadas erróneamente como `bad`.
- **153 instancias** de la clase `bad` fueron clasificadas correctamente, mientras que **147** fueron clasificadas erróneamente como `good`.

Interpretación del Resultado del Proyecto

En el ejemplo que proporcionas con la interfaz gráfica, el modelo predice una probabilidad del **99.90%** de que el cliente pertenece a la clase `good`. Esto se basa en los parámetros seleccionados para la instancia específica, como:

- Estado de la cuenta: **0 a 200 unidades**.
- Duración: **10 meses**.
- Propósito del crédito: **Muebles/equipos**.
- Estado civil: **Hombre soltero**.
- Propiedad: **Coche**.

El alto porcentaje refleja que el modelo identifica esta instancia como similar a otras clasificadas como `good` en el conjunto de datos de entrenamiento.

Observaciones

1. El modelo Lazy-IBK es adecuado para este problema debido a su simplicidad y capacidad para adaptarse a datos con fronteras claras.
2. Aunque la precisión general es del 72%, el desempeño para la clase `bad` es inferior, lo que podría indicar la necesidad de:
    a. Aumentar el valor de K para suavizar las decisiones del modelo.
    b. Probar diferentes métricas de distancia (por ejemplo, Manhattan o Chebyshev).
    c. Balancear las clases para mejorar el reconocimiento de la clase `bad`.
3. Los resultados pueden ser utilizados para decisiones iniciales de riesgo crediticio, pero el modelo podría mejorarse para aplicaciones más críticas.

Explicación por que se escogió este algoritmo

El algoritmo **Lazy-IBK** es ideal para este análisis porque es simple, no hace suposiciones sobre la distribución de los datos y clasifica basándose en las similitudes entre instancias. Esto lo hace adecuado para conjuntos de datos como el de crédito bancario, donde las relaciones entre atributos son complejas y no lineales. Su capacidad para identificar patrones locales y manejar tanto variables categóricas como numéricas lo convierte en una herramienta efectiva para evaluar perfiles crediticios y tomar decisiones basadas en datos históricos de manera interpretativa y confiable.

5. Algoritmo elegido para la interpretación correcta de los datos

El algoritmo elegido es **FUNCTION LOGISTIC,** en weka debido a su capacidad para modelar problemas de clasificación binaria, como distinguir entre “good” y “bad”. Con una base estadística sólida. Este método tiene varias ventajas y características que lo hacen adecuado para analizar datasets como **credit.g.arff**, que contiene variables categóricas y continúas relacionadas con la evaluación de riesgos crediticios.

Ventajas que tiene Logistic Regression.

- **Modelado Probabilístico.**
    - La regresión logística no solo clasifica las instancias como "good" o "bad", sino que también proporciona probabilidades asociadas a cada predicción.
    - Esto es crucial en aplicaciones como el análisis crediticio, donde no solo se busca saber si un cliente es riesgoso, sino también cuán seguro o inseguro es prestarles.

- **Interpretabilidad.**
    - Los coeficientes en el modelo representan la contribución de cada atributo a la probabilidad de una clase específica.
    - Ejemplo: En el modelo generado, un coeficiente positivo para foreign_worker=no indica que ser un trabajador extranjero incrementa significativamente la probabilidad de ser clasificado como "bad" credit.
    - Los odds ratios permiten interpretar fácilmente el impacto de cada variable: si el ratio es mayor a 1, incrementa la probabilidad de la clase "good"; si es menor a 1, reduce dicha probabilidad.

- **Manejo de Variables Categóricas.**
    - Logistic Regression en Weka maneja variables categóricas a través de la codificación dummy, como se observa en atributos como checking_status o purpose.
    - Esta característica es ideal para el dataset german_credit, que contiene múltiples variables categóricas como housing, personal_status, y job.

- **Generalización Controlada.**
    - El uso del término de regularización (ridge parameter -R) ayuda a prevenir el sobreajuste, especialmente en datasets con muchas variables. En tu caso, se especificó un valor extremadamente pequeño (1.0E-8), lo que significa que el modelo confía principalmente en los datos, pero está preparado para evitar coeficientes extremadamente grandes.

Ventajas de Logistic Regression sobre Otros Algoritmos en WEKA.

- **Comparación con J48 (Árbol de Decisión).**
    - **Ventaja clave de Logistic:** Produce un modelo matemático más robusto que generaliza mejor en problemas con relaciones lineales o casi lineales entre los atributos y las clases.
    - J48 puede ser menos interpretable en datasets con muchas variables categóricas debido a la complejidad del árbol.
    - Logistic permite interpretar el peso exacto de cada atributo.

- **Comparación con Naive Bayes.**
    - **Ventaja clave de Logistic:** Considera interacciones entre atributos, mientras que Naive Bayes asume independencia condicional entre ellos.
    - En el dataset german_credit, algunos atributos como credit_amount y duration están correlacionados, lo que hace que Logistic sea más apropiado.

- **Comparación con SVM (Máquinas de Soporte Vectorial).**
    - **Ventaja clave de Logistic:** Es más interpretable. Mientras que SVM genera una frontera de decisión óptima, no proporciona coeficientes que expliquen cómo cada atributo contribuye a la clasificación.
    - Logistic también es más rápido de entrenar en datasets con menos complejidad.

Interpretación de los Resultados del Modelo.

Coeficientes y Odds Ratios

- **Coeficientes:** Indican la dirección y magnitud del impacto de cada atributo en la clase objetivo ("good" o "bad").
    - Por ejemplo, un coeficiente positivo para checking_status=no checking (0.9339) significa que no tener cuenta corriente aumenta la probabilidad de ser "good".
    - El valor negativo de credit_amount (-0.0001) sugiere que montos de crédito más altos están asociados con mayor riesgo.

- **Odds Ratios**: Son exponentes de los coeficientes y facilitan la interpretación en términos de proporciones. Un valor mayor a 1 indica un impacto positivo, mientras que valores menores a 1 indican un impacto negativo.

Matriz de Confusión.

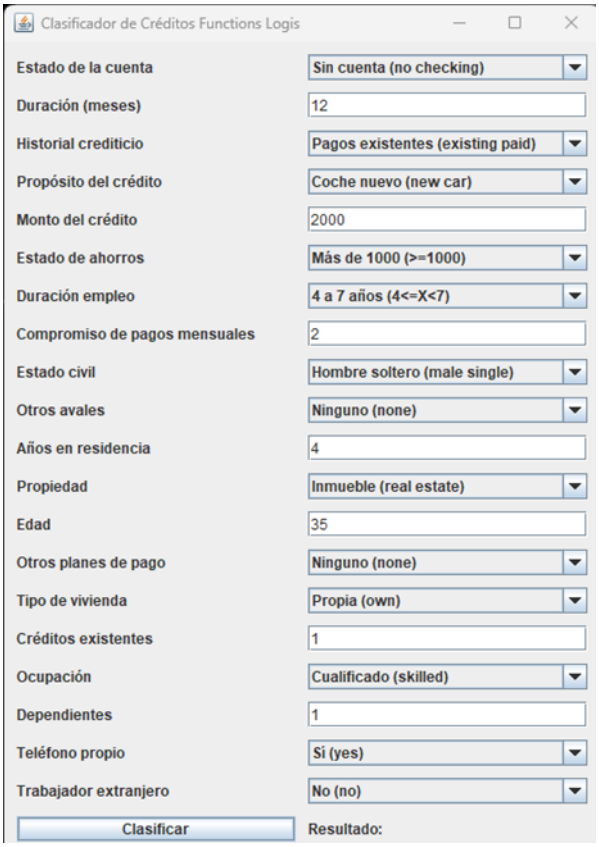
Los valores en la matriz indican:

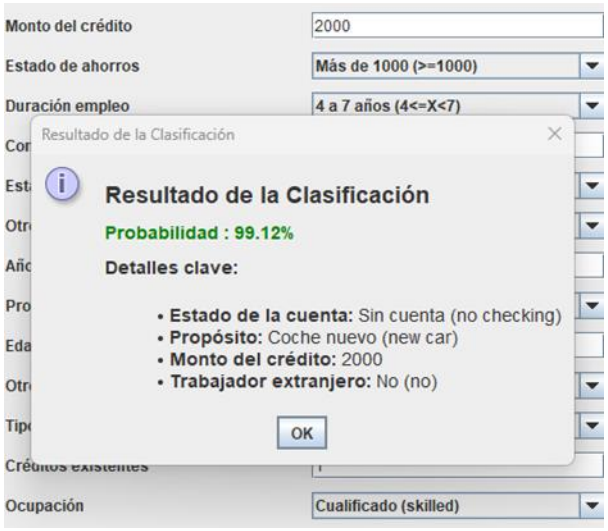
- **605 instancias** de "good" se clasificaron correctamente.
- **147 instancias** de "bad" se clasificaron correctamente.
- El resto son errores de clasificación.

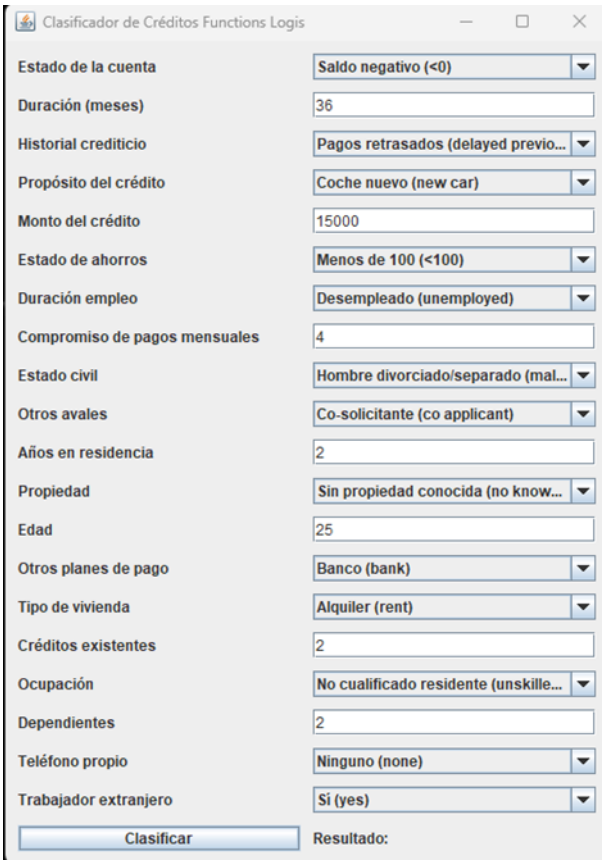
Métrica de Evaluación

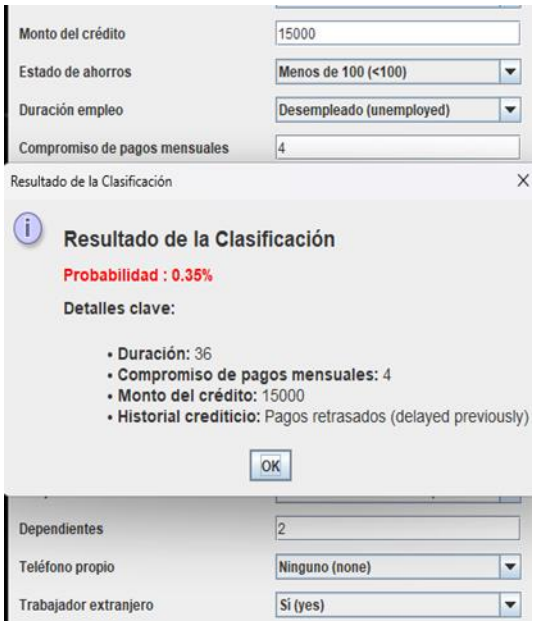
- **Accuracy (75.2%):** Indica que el modelo clasifica correctamente 3 de cada 4 instancias.
- **Precisión:**
    - Para "good" (0.798): De todas las instancias predichas como "good", el 79.8% son correctas.
    - Para "bad" (0.607): De todas las instancias predichas como "bad", el 60.7% son correctas.
- **ROC Area (0.785):** Una puntuación razonable que indica que el modelo tiene buena capacidad de discriminación.

Elegir Logistic Regression para analizar el dataset **credit.g.arff** fue una decisión fundamentada en su capacidad para manejar problemas de clasificación binaria con datos mixtos, proporcionar interpretaciones detalladas y ser computacionalmente eficiente. Su capacidad para cuantificar el impacto de cada variable y modelar probabilidades lo hace ideal para evaluar riesgos crediticios.






Clasificador de Créditos Functions Logis
Estado de la cuenta
Saldo negativo (<0)
Duración (meses)
36
Historial crediticio
Pagos retrasados (delayed previo...
Propósito del crédito
Coche nuevo (new car)
Monto del crédito
15000
Estado de ahorros
Menos de 100 (<100)
Duración empleo
Desempleado (unemployed)
Compromiso de pagos mensuales
4
Estado civil
Hombre divorciado/separado (mal...
Otros avales
Co-solicitante (co applicant)
Años en residencia
2
Propiedad
Sin propiedad conocida (no know...
Edad
25
Otros planes de pago
Banco (bank)
Tipo de vivienda
Alquiler (rent)
Créditos existentes
2
Ocupación
No cualificado residente (unskille...
Dependientes
2
Teléfono propio
Ninguno (none)
Trabajador extranjero
Sí (yes)
Clasificar
Resultado:


Monto del crédito
15000
Estado de ahorros
Menos de 100 (<100)
Duración empleo
Desempleado (unemployed)
Compromiso de pagos mensuales
4
Resultado de la Clasificación
Resultado de la Clasificación
Probabilidad : 0.35%
Detalles clave:
• Duración: 36
• Compromiso de pagos mensuales: 4
• Monto del crédito: 15000
• Historial crediticio: Pagos retrasados (delayed previously)
OK
Dependientes
2
Teléfono propio
Ninguno (none)
Trabajador extranjero
Sí (yes)