Artículo Original Pág. 47 - 60 ISSN 2415-2323

ILUMATE VOLUMEN 12

ANÁLISIS DE RIESGO PARA PRÉSTAMOS BANCARIOS

Risk analysis for bank loans

Juan Salvador Vilca Paredes
slvdrvlc@gmail.com
Instituto de Investigación en Ciencia y Tecnología,
Universidad La Salle - Bolivia

Resumen

En estos tiempos de pandemia, existen muchas dificultades a la hora de saber a quién prestar, inclusive si el cliente es una persona que demuestra solvencia. El objetivo es identificar si es prudente o no realizar un préstamo a los clientes según el monto del préstamo que solicitan y los datos que se tienen de los mismos. Es un problema de clasificación para predecir si es recomendable o no un préstamo. Se debe predecir valores discretos basados en un conjunto dado de variables independientes. Se empleo la metodología cuantitativa con diseño de investigación no experimental en el tipo de estudio correlacional. Los resultados fueron que no siempre las hipótesis pueden estar alineadas con el análisis y evaluación de los datos, pues los resultados demostraron que los solicitantes con más ingresos no necesariamente eran los que tenían mayor posibilidad de acceder al préstamo.

Palabras claves

Aprendizaje automático, ciencia de datos, clasificación, python, regresión logística

Abstract

In these times of pandemic, there are many difficulties when it comes to knowing who to lend to, even if the client is a person who demonstrates solvency. The objective is to identify whether or not it is prudent to make a loan to clients based on the amount of the loan they request and the data they have about them. It is a classification problem to predict whether or not a loan is recommended. Discrete values must be predicted based on a given set of independent variables. The quantitative methodology was used with a non-experimental research design in the correlational type of study. The results were that the hypotheses could not always be aligned with the analysis and evaluation of the data, since the results showed that the applicants with the highest income were not necessarily the ones with the greatest possibility of accessing the loan

Key words

Machine learning, data science, classification, python, logistic regression

Introducción

La predicción de préstamos es un problema muy común de la vida real que cada banco enfrenta. Si se hace correctamente, puede ahorrar muchas horas de trabajo. Es importante tener en cuenta que las predicciones por lo general son de apoyo para la toma de decisiones, es decir, que apoyan a profesionales a tomar decisiones en sus áreas y no así reemplazarlos en sus tareas.

El trabajo consiste en ayudar a una empresa bancaria que realiza préstamos con presencia en todas las áreas urbanas, semiurbanas y rurales. El cliente primero solicita un préstamo hipotecario después de que la compañía valida la elegibilidad del cliente para el préstamo. Sin embargo, hacer esto manualmente lleva mucho tiempo. Se propone realizar la predicción de la aprobación de los clientes para el préstamo ayudando de esta manera a respaldar la elegibilidad del préstamo en función de la información del cliente.

Se utilizó el modelo de regresión logística para realizar una clasificación binaria donde las predicciones fueron:

Y/1(si) para indicar que el préstamo es recomendable.
N/0(no) para indicar que el préstamo no es recomendable.

Referentes conceptuales.

La Regresión Logística según Amat (2016), desarrollada por David Cox en 1958, es un método de regresión que permite estimar la probabilidad de una variable cualitativa binaria en función de una variable cuantitativa.

Es un problema de clasificación en el que tenemos que predecir si se aprobase o no un préstamo. En un problema de clasificación, tenemos que predecir valores discretos basados en un conjunto dado de variables independientes, la clasificación puede ser de dos tipos:

Clasificación binaria: en esta clasificación tenemos que predecir cualquiera de las dos clases dadas. Por ejemplo: clasificar el género como masculino o femenino, predecir el resultado como ganar o perder, etc., o como en este paso aceptar o rechazar la solicitud.

Clasificación multiclase: Aquí tenemos que clasificar los datos en tres o más clases. Por ejemplo: clasificar el género de una película como comedia, acción o romántico, clasificar frutas como naranjas, manzanas o peras, etc.

> Según Amat (2016):
> "Una de las principales aplicaciones de la regresión logística es la de clasificación binaria, en el que las observaciones se clasifican en un grupo u otro dependiendo del valor que tome la variable empleada como predictor. Por ejemplo, clasificar a un individuo desconocido como hombre o mujer en función del tamaño de la mandíbula" (p.3).

Para la predicción de la variable objetivo se aplica el modelo de regresión

logística para predecir el resultado binario.

Según Fernandez (2011):
"Con el fin de estimar **β** = β_1 β_2 L βk y analizar el comportamiento del modelo estimado se toma una muestra aleatoria de tamaño n dada por ($X_{i,}$ $Y_{i)\,i}$ = $_1$, $_2$, ...,$_n$ donde el valor de las variables independientes es X_i = ($X_{i1,}$ X_{i2} , ... , X_{ik}) e Y_i ∈ [0.1] es el valor observado de Y en el i-ésimo elemento de la muestra" (p.4)

La regresión logística, a pesar de su nombre, es un modelo lineal para clasificación en lugar de regresión. La regresión logística también se conoce en la literatura como regresión logit, clasificación de máxima entropía (MaxEnt) o clasificador log-lineal. En este modelo, las probabilidades que describen los posibles resultados de un solo ensayo se modelan utilizando una función logística.

Métodos.

Se emplean las siguientes librerías de Python:

- pandas
- numpy
- seaborn
- matplotlib
- sklearn

Pandas es una de las librerías de python más útiles para los científicos de datos. Las estructuras de datos principales en pandas son Series para datos en una dimensión y DataFrame para datos en dos dimensiones.

Estas son las estructuras de datos más usadas en muchos campos tales como finanzas, estadística, ciencias sociales y muchas áreas de ingeniería. Pandas destaca por lo fácil y flexible que hace la manipulación de datos y el análisis de datos.

NumPy proporciona una estructura de datos universal que posibilita el análisis de datos y el intercambio de datos entre distintos algoritmos. Las estructuras de datos que implementa son vectores multidimensionales y matrices con capacidad para gran cantidad de datos.
Además, esta librería proporciona funciones matemáticas de alto nivel que operan en estas estructuras de datos.

Seaborn es una librería gráfica basada en matplotlib, especializada en la visualización de datos estadísticos. Se caracteriza por ofrecer un interfaz de alto nivel para crear gráficos estadísticos visualmente atractivos e informativos.

Seaborn considera la visualización como un aspecto fundamental a la hora de explorar y entender los datos. Se integra muy bien con la librería de manipulación de datos pandas.

Matplotlib es la librería gráfica de python estándar y la más conocida. Se puede usar matplotlib para generar gráficos de calidad necesaria para publicarlas tanto en papel como digitalmente.

Con matplotlib se puede crear muchos tipos de gráficos: series temporales, histogramas, espectros de potencia, diagramas de barras, diagramas de errores, etc.

Scikit-learn es una librería de python para Machine Learning y Análisis de Datos. Está basada en NumPy, SciPy y Matplotlib. Las ventajas principales de scikit-learn son su facilidad de uso y la gran cantidad de técnicas de aprendizaje automático que implementa.

Con scikit-learn se puede realizar aprendizaje supervisado y no supervisado. Podemos usarlo para resolver problemas tanto de clasificación y como de regresión.

Es muy fácil de usar y aprender porque tiene una interfaz simple y muy

consistente. Se puede verificar que el interfaz es consistente cuando es posible cambiar de técnica de machine learning cambiando sólo una línea de código.

Otro punto a favor de scikit-learn es que los valores de los hiper-parámetros tienen unos valores por defecto adecuados para la mayoría de los casos.
Es una de las herramientas más eficientes que contiene muchas funciones incorporadas que se pueden usar para modelar en Python.

Específicamente se utiliza la clase sklearn.linear_model. Logistic Regression para implementar el modelo, el cual tiene los siguientes métodos:

Fig. 1 Métodos de la clase LogisticRegression

`decision_function`(X)	Predecir puntuaciones de confianza para muestras.
`densify`()	Convierta la matriz de coeficientes a un formato de matriz densa.
`fit`(X, y [, sample_weight])	Ajuste el modelo de acuerdo con los datos de entrenamiento proporcionados.
`get_params`([profundo])	Obtenga parámetros para este estimador.
`predict`(X)	Predecir etiquetas de clase para muestras en X.
`predict_log_proba`(X)	Predecir el logaritmo de las estimaciones de probabilidad.
`predict_proba`(X)	Estimaciones de probabilidad.
`score`(X, y [, sample_weight])	Devuelve la precisión media en las etiquetas y los datos de prueba dados.
`set_params`(** params)	Establezca los parámetros de este estimador.
`sparsify`()	Convierta la matriz de coeficientes a formato disperso.

Fuente: Elaborado por: Sklearn

También es importante mencionar que las pruebas fueron realizadas en Google Colab.

Colab es un servicio en la nube, basado en los Notebooks de Jupyter, que permite el uso gratuito de las GPUs y TPUs de Google, con librerías como: Scikit-learn, PyTorch, TensorFlow, Keras y OpenCV. Con Python 2.7 y 3.6, aún no está disponible para R y Scala.

Aunque tiene algunas limitaciones, es una herramienta ideal, no solo para

practicar y mejorar nuestros conocimientos en técnicas y herramientas de Data Science y Machine Learning, sino también para el desarrollo de aplicaciones de deep learning, sin tener que invertir en recursos hardware o en la nube.

Con Colab se pueden crear notebooks o importar los que ya tengamos creados, además de compartirlos y exportarlos cuando queramos. Esta fluidez a la hora de manejar la información también es aplicable a las fuentes de datos que usemos en nuestros proyectos (notebooks), de modo que podremos trabajar con información contenida en nuestro propio Google Drive, unidad de almacenamiento local, github e incluso en otros sistemas de almacenamiento en la nube.

Resultados y discusión

Hipótesis

Es muy útil e importante desarrollar algunas hipótesis en trabajos de aprendizaje automático, es necesario entender el problema a detalle entendiendo el enunciado del problema a fondo y antes de mirar los datos, con una lluvia de ideas de tantos factores como sea posible que puedan afectar el resultado

Salario: los solicitantes con altos ingresos deberían tener más posibilidades de aprobación del préstamo.

Historial previo: los solicitantes que hayan pagado sus deudas anteriores deberían tener mayores posibilidades de aprobación del préstamo.

Monto del préstamo: la aprobación del préstamo también debe depender del monto del préstamo. Si el monto del préstamo es menor, las posibilidades de aprobación del préstamo deben ser altas.

Plazo del préstamo: el préstamo por un período de tiempo y una cantidad

menores debería tener mayores posibilidades de aprobación.
EMI (pago mensual estimado/cuota mensual): Cuanto menor sea la cantidad para pagar mensualmente para reembolsar el préstamo, mayores serán las posibilidades de aprobación del préstamo.

El dataset tiene las siguientes características:

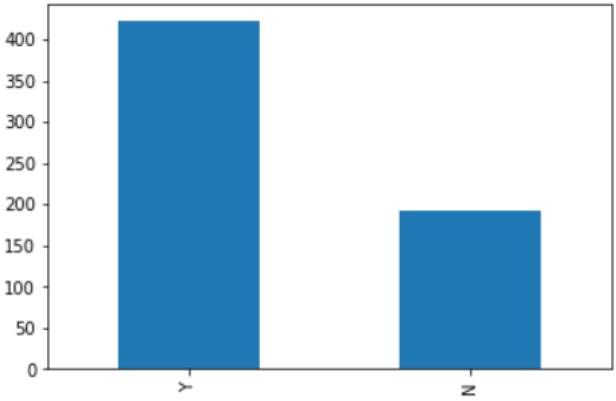
Loan_ID: ID de préstamo único
Gender: Genero (Masculino / Femenino)
Married: Estado civil (S / N)
Dependents: número de dependientes (0, 1, 2, 3+)
Education: Educación del solicitante (graduado / no graduado)
Self_Employed: Trabajadores por cuenta propia (S / N)
ApplicantIncome: Ingresos del solicitante
CoapplicantIncome: Ingresos del garante
LoanAmount: Monto del préstamo en miles de dólares
Loan_Amount_Term: Plazo del préstamo
Credit_History: Historial crediticio (S / N)
Property_Area: área de ubicación del inmueble (Urbano / Semiurbano / Rural)
Loan_Status: Estado del préstamo (S / N) esta es la variable objetivo

Las características de las variables del dataset son:

- Características categóricas: estas características tienen categorías (género, casado, autónomo, historial crediticio, estado del préstamo).
- Características ordinales: variables en características categóricas que tienen algún orden (dependientes, educación, área de propiedad).
- Características numéricas: estas características tienen valores numéricos (Ingreso del solicitante, Ingreso del garante, Monto del prestamo, Plazo del préstamo).

Primero se realiza un análisis unitario de las variables, empezando por la variable objetivo Loan_Status que es categórica.

Fig. 2 Tabla de solicitantes de préstamo aprobados y rechazados

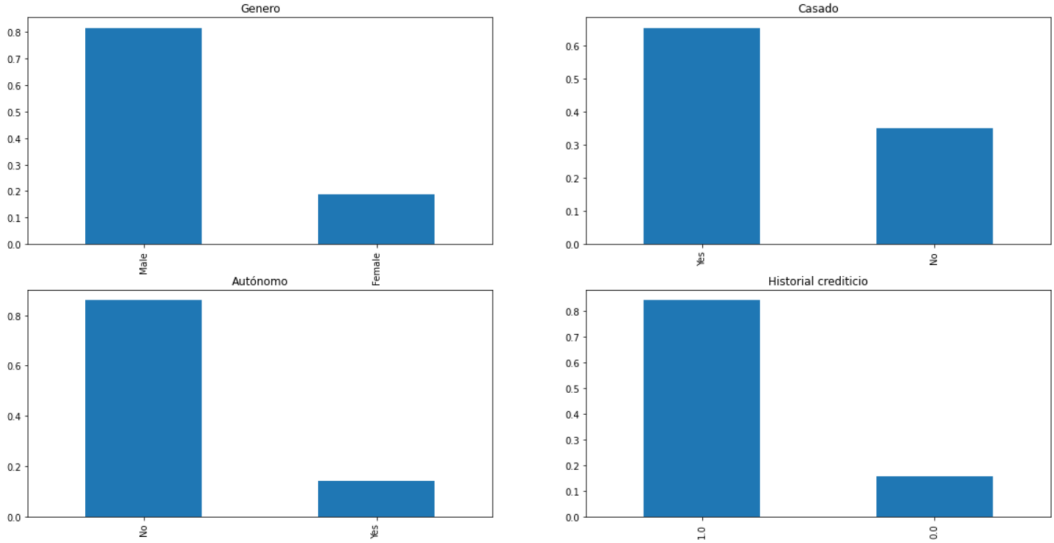


Fuente: Elaboración propia

Donde se puede observar que, casi el doble de solicitantes es aprobado para el crédito.

Analizando las variables independientes categóricas se tiene:

Fig.3 Tablas de Genero, Casado, Autónomo e Historial crediticio del solicitante



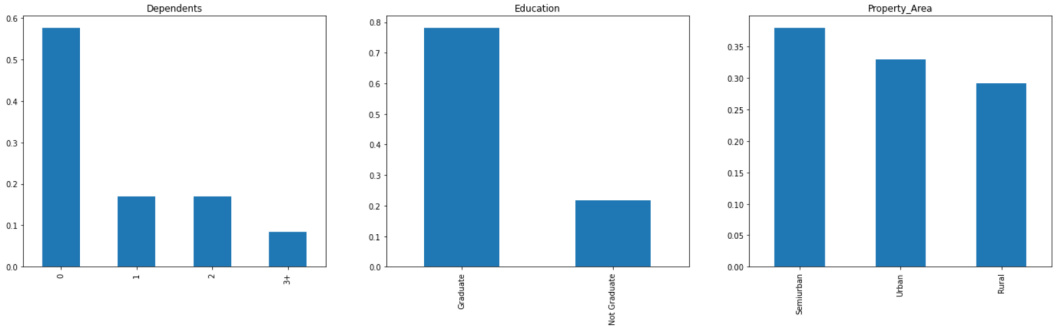
Fuente: Elaboración propia

Se puede ver en los gráficos de barras anteriores que:

- El 80% de los solicitantes son hombres.
- Alrededor del 65% de los solicitantes están casados.
- Alrededor del 15% de los solicitantes son autónomos.
- Alrededor del 85% de los solicitantes tienen historial de crédito.

Analizando las variables independientes ordinales se tiene:

Fig. 4 Tabla de Dependientes, Educación y área de la propiedad del solicitante



Fuente: Elaboración propia

- La mayoría de los solicitantes no tienen dependientes.
- Alrededor del 80% de los solicitantes son graduados.
- La mayoría de los solicitantes son del área semiurbana.

Analizando las variables independientes numéricas que tiene:

- La mayoría de los datos en la distribución de los ingresos de los solicitantes están hacia la izquierda, lo que significa que no se distribuye normalmente.
- El diagrama de caja confirma la presencia de muchos valores atípicos / extremos.
- Lo más probable es que sea por la disparidad de ingresos en la sociedad.
- Con el hecho de que estamos mirando a persona con diferentes niveles educativos.

Analizando la distribución de la variable LoanAmount se tiene:

- Se observa muchos valores atípicos en el diagrama de caja de esta variable.
- Pero la distribución es bastante normal.

El Historial crediticio contra estado de préstamo refleja:

- Que las personas con historial crediticio tienen más probabilidades de obtener la aprobación de sus préstamos visto de otra manera al solicitar un préstamo la primera vez es difícil.

Vemos la correlación entre todas las variables numéricas donde:

- Vemos que las variables más correlacionadas son Ingreso del solicitante con monto del préstamo e historial crediticio con estado del préstamo.

En el tratamiento del dataset:

- Hay menos valores perdidos en las características Sexo, Casado, Dependientes, Historial de crédito y Autónomo, por lo que podemos completarlos usando la moda de las características.
- Usamos la mediana para completar los valores nulos, ya que antes vimos que el monto del préstamo tiene valores atípicos, por lo que la media no será el enfoque adecuado, ya que se ve muy afectada por la presencia de valores atípicos.
- Para manejar variables categóricas, existen otros métodos. como necesitamos convertir todas las variables categóricas en numéricas, se utilizará el método get_dummies.

En la construcción del modelo:

Sklearn requiere la variable de destino en un conjunto de datos separado por lo que eliminamos nuestra variable de destino del conjunto de datos y la guardamos en otro conjunto de datos.

Después se crean variables ficticias para las variables categóricas. La variable ficticia convierte las variables categóricas en una serie de 0 y 1, lo que las hace mucho más fáciles de cuantificar y comparar.

Se entrena el modelo en el conjunto de datos de entrenamiento y se harán predicciones para el conjunto de datos de prueba. Una forma de hacerlo es

dividir el conjunto de datos en dos partes: entrenamiento y validación.

Usando la función train_test_split de sklearn para dividir el conjunto de datos. Con LogisticRegression, fit () y precision_score de sklearn ajustamos el modelo de regresión logística.

Se debe predecir Loan_Status y calcular su precisión con el método predict () y accuracy_score ().

Donde las predicciones son casi un 80% precisas, es decir, se logró identificado correctamente el 80% del estado del préstamo.

Metodología:

Se empleo la metodología cuantitativa con diseño de investigación no experimental en el tipo de estudio correlacional.

Conclusiones:

Hoy en día, el negocio de préstamos se vuelve cada vez más popular y muchas personas solicitan préstamos por diversas razones. Sin embargo, hay casos en los que las personas no devuelven la mayor parte del monto del préstamo al banco, lo que resulta en una gran pérdida financiera. Si se logra clasificar de manera eficiente a los solicitantes por adelantado, evitaría en gran medida la pérdida financiera.

En este estudio, primero se limpió el conjunto de datos y se realizó el análisis de datos exploratorios. Se cubrieron las estrategias para abordar tanto los valores perdidos como los conjuntos de datos desequilibrados.

A partir del análisis exploratorio de datos y evaluación individual de las variables, se puede generar información a partir de los datos y entender cómo se relaciona cada una de las variables individuales con la variable objetivo. También se puede notar que la librería de sklearn nos facilita el

trabajo enormemente al permitir trabajar con hiper parámetros por defecto que por lo general son los más adecuados y en este caso se pudo obtener un resultado aceptable.

Se pudo evaluar que no siempre las hipótesis pueden estar alineadas con el análisis y evaluación de los datos, por ejemplo, en este caso los solicitantes con más ingresos no necesariamente eran los que tenían mayor posibilidad de acceder al préstamo, por otro lado, la variable que más impacto tenía era el historial crediticio del solicitante.

Referencias.

- Amat, J. (2016, 8). *Regresion logistica simple y multiple.* Retrieved from ciencia de datos: https://www.cienciadedatos.net/documentos/27_regresion_logistica_simple_y_multiple
- Borrajo, D. (2006). *Aprendizaje autom´atico.* Madrid.
- Fernandez, S. d. (2011). *Regresión logística.* Madrid.
- Garreta, R. (2013). Learning scikit-learn: Machine Learning in Python.
- Mitchel, T. (1997). Machine Learning. McGraw Hill.
- Murphy, K. (2012). *Machine Learning: A probabilistic perspective.* The MIT Press.
- Sklearn. (2020, 12 2). *Logistic Regression.* Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Artículo recibido: 22-09-2020
Artículo aceptado: 05-11-2020