

Aprendizaje Automático para Análisis de Datos: Análisis y Reporte sobre el desempeño del modelo

Primer modelo de regresión lineal (Implementado por Guillermo Cepeda)

Guillermo Romeo Cepeda Medina

A01284015

En el siguiente texto se incluye un enlace a la implementación del modelo implementado por Guillermo Romeo Cepeda Medina que será analizado estadísticamente en esta entrega:

https://github.com/Memo9494/Analisis_a01284015_IA

Descripción

En este documento se realiza un análisis acerca de los modelos de inteligencia artificial implementados en entregas pasadas, con el objetivo de evaluar el modelo y obtener conclusiones del mismo.

Nombre del dataset: Fake Bills

Link: <https://www.kaggle.com/datasets/alexandrepetit881234/fake-bills>

Descripción del dataset:

El dataset incluye 1000 billetes reales y 500 falsos

Así como las siguientes clases de características de los billetes:

is_genuine: boolean diagonal: float height_left: float height_right: float margin_low: float margin_upper: float length: float

Todos los valores de las clases son de tipo float (Decimales con poca precisión), excepto la característica de si es falso o no, que es booleana.

El modelo que estamos implementando intenta resolver un problema de clasificación, queremos saber si los billetes son falsos o no.

Como es documentado en el repositorio de implementación, se realiza una prueba tanto con los datos de entrenamiento, como de prueba:

A continuación los resultados de probar el modelo con ambas poblaciones:

Datos prueba:

Falso positivo: 3

Falso negativo: 0

Verdadero positivo: 197

Verdadero negativo: 93

Exactitud: 0.9897610921501706

Precisión: 0.985

Exhaustividad: 1.0

F1: 0.9924433249370277

Datos entrenamiento:
Falso positivo: 45
Falso negativo: 1
Verdadero positivo: 970
Verdadero negativo: 447
Exactitud: 0.9685577580314423
Precisión: 0.9556650246305419
Exhaustividad: 0.9989701338825953
F1: 0.9768378650553877
Datos prueba:

Separación de las variables para el uso de ambos modelos (Realizado por Guillermo Cepeda)

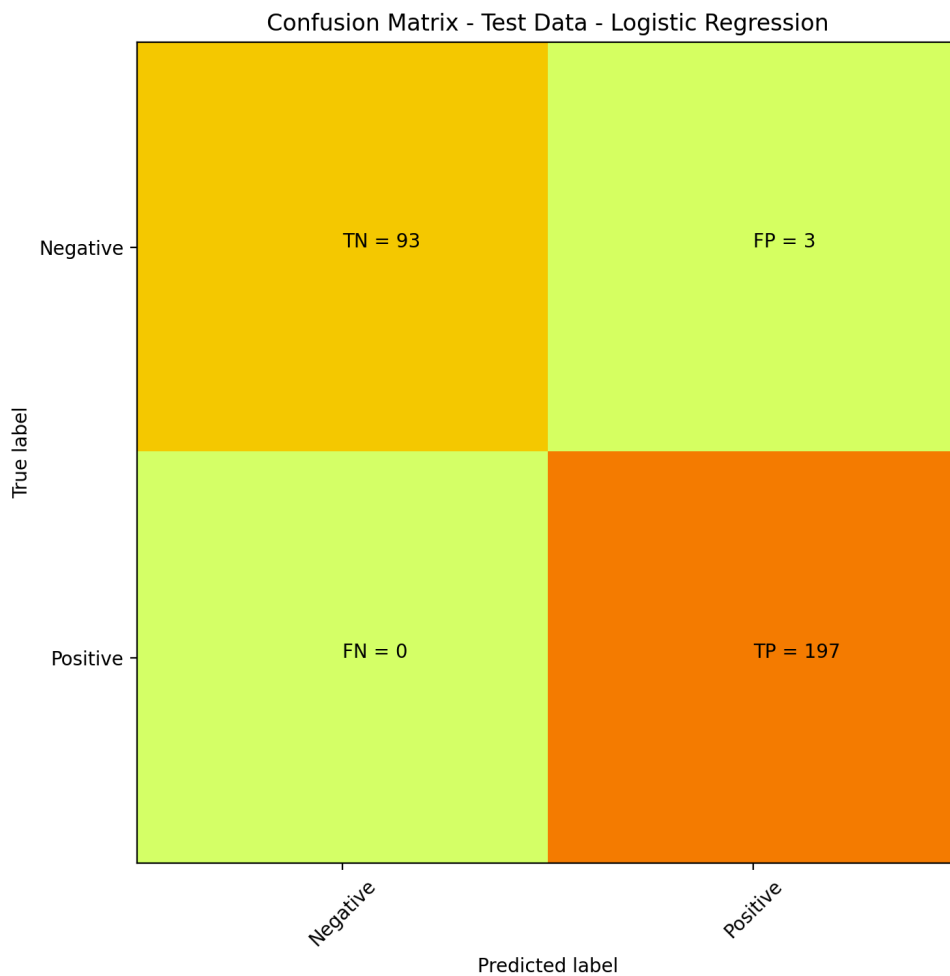
Para empezar con el análisis del modelo se separaron dos partes del dataset, una para entrenamiento y la otra para hacer la predicción. Estas se separaron aleatoriamente para minimizar el sesgo de ambas poblaciones

```
# Dividir los datos en entrenamiento y prueba x sólo las variables que se
consideran importantes length, margin_low y margin_up
X = df[['length', 'margin_low', 'margin_up']]

# X = df[['diagonal', 'height_left', 'height_right', 'margin_low', 'margin_up',
'length']]
y = df['is_genuine']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

Análisis de sesgo

Esto nos permitió poder utilizar nuestro modelo sin la preocupación de un sesgo dentro del mismo dataset, no obstante el modelo de predicción realizado por Guillermo Cepeda sí cuenta con un sesgo, esto se representa en la siguiente gráfica:



Se puede apreciar, que aunque pequeño, se encuentra un sesgo en los positivos, puesto que el modelo realizó 3 falsos positivos, por lo que tiende a interpretar los datos con un sesgo a que sean positivos.

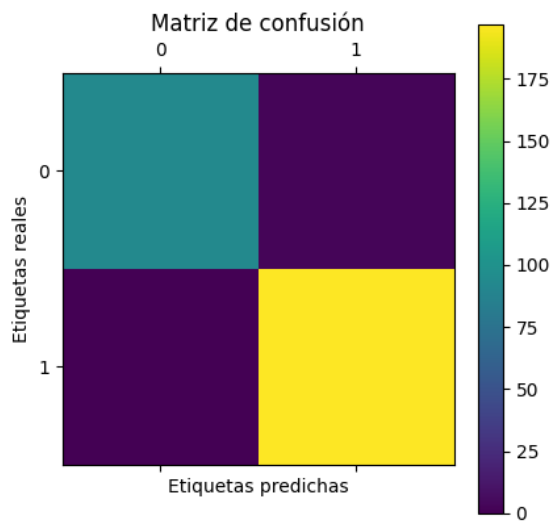
Otros datos que nos hablan del sesgo en un modelo de clasificación binaria:

Exactitud: 0.9897610921501706

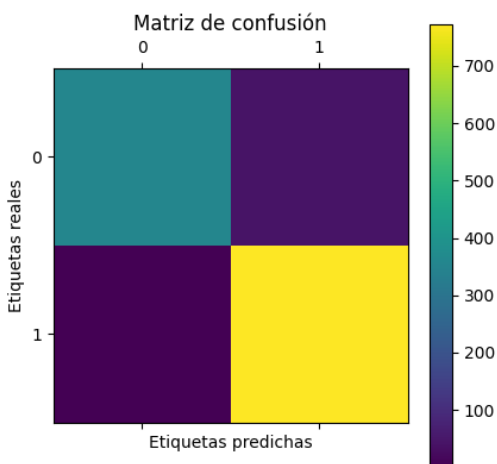
Precisión: 0.985

Exhaustividad: 1.0

Aquí la exactitud nos habla de que tan desbalanceados están los datos, el valor es muy cercano al 1 por lo que podemos concluir que no hay sesgo o es muy poco.



Datos entrenamiento:



Esta idea se puede retomar usando los datos de entrenamiento, los cuales confirman un sesgo a los positivos, al tener un dataset mayor se puede apreciar que se conserva el sesgo con una poblacion mayor, el sesgo es proporcional, pero no lo quita de ahi.

Análisis de la varianza

En cuánto a la varianza se refiere, se busca cuantificar la precisión, que tan bueno es el modelo para predecir una muestra como positiva, y que tan bueno es para predecir la clase. Por lo que la métrica F1-score

F1: 0.9924433249370277

Por lo que podemos concluir que la varianza es realmente poca en el modelo, con sólo 3 falsos positivos.

Los cambios que realicé para alcanzar estos resultados en mis predicciones se centraron en la normalización de los datos, así como la reducción de las variables. Esto para optimizar el

proceso de entrenamiento y conseguir un aprendizaje más eficiente. No obstante el no incluir variables y perder datos de significancia al transformarlos entre 0 y 1. Además de que las iteraciones del modelo no son muchas. Utilizar el modelo con los datos tal cual y aumentar el número de iteraciones podrían contribuir a obtener un mejor resultado en nuestra predicción, esto es importante dado que estamos hablando de billetes, los cuáles tienen muchísimo valor.

Análisis overfitted

Para analizar si mi modelo está overfitted, realicé la hipótesis de mis modelos de entrenamiento también y conseguí ningún error, lo cual era de esperarse debido a que eran los datos de prueba. Sin embargo me parece que el modelo está overfitted debido a los falsos positivos que resultados de mis modelos de test. A pesar de que el rendimiento es muy parecido en ambas poblaciones, tengo entendido que un valor alto de F1: es un indicador grande de que un modelo es overfitted. No obstante, también es de considerar que los datos que descargué de kaggle no tienen una fuente de dónde vienen, por lo que puede que estos datos no sean reales pero y que sea un dataset especial para hacer este tipo de pruebas y que los resultados deseados sean de este tipo.